UNIVERSIDAD CARLOS III DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

INGENIERÍA DE TELECOMUNICACIÓN

PROYECTO DE FIN DE CARRERA

SET-TOP-BOX VIRTUAL EN LA NUBE

Autor: David de la Horra Iglesias

Tutor: Daniel Díaz Sánchez

Agradecimientos

A mis padres y a mi hermana por soportarme estos largos años de carrera, cuando volvía de la universidad sin ganas de hablar con nadie o enfadado tras suspender un examen siempre estabais ahí. En el fondo echare de menos el "ponte a estudiar".

A Álvaro por estar ahí para lo que necesitara y ser mi seguro de fiesta todos los findes.

A mis compañeros de la Uni en especial a Ana "mi compi", Miguelin, Alcolea, Adri, Victor, Javi, Cris, Palo y todos aquellos que me habéis soportado estos años y que habéis hecho ameno el paso por la universidad.

A todos mis compañeros de basket especialmente a Rafa, IL, Soti, Karel, Busta, Chuster, Carlitos, Pablo, Bardi, Roso, Fon, Chus, Goyo, AM, CAE y muchos otros que seguro que se me están olvidando, por todos los buenos que quedarán en mi memoria y por ser mi vía de escape de la universidad.

A mi familia (al resto) que aunque os haya visto menos de lo que debería, por estar siempre ahí.

A mis amigos de la playa que aunque os veo menos de lo que me gustaría, por hacerme reír y conseguir que desconecte en las vacaciones de los estudios.

A mi peña del pueblo el revolcón, por esos buenos San Roques y fiestas que he pasado con vosotros.

A mi tutor, Dani gracias por tu confianza y tu tiempo, sin ella ahora mismo esto no estaría escribiendo esto.

A todos los que estáis leyendo esto y estáis indignados de no aparecer, se que alguno se me habrá olvidado, y en general a todos los que a lo largo de estos años habéis pasado por mi vida más o menos tiempo y habéis conseguido que hayan sido unos años fantásticos.

Resumen

Actualmente los servicios de televisión sobre IP están encontrando una gran aceptación en el mercado. Esto es debido a su fácil implantación y al crecimiento del ancho de banda que llega a los hogares, de forma que se pueda proporcionar una emisión fluida del contenido. Los proveedores de IPTV son operadores de banda ancha, que aprovechan la infraestructura para reservar un ancho de banda para la televisión. Estas operadoras han integrado los servicios de voz, datos y televisión ayudados por los protocolos de comunicación de las redes IP. A esta integración se la ha denominado "Triple Play".

La IPTV permite reproducir canales de televisión en directo, canales de televisión en diferido y video a la carta. Lo único necesario es un Set-Top-Box (STB) que decodificará la señal y proporcionará la seguridad necesaria. Entorno a la estrategia de prestación de servicio, hay varias tendencias. Por una parte están las soluciones que usan los estándares como DVB, OIPF o TISPAN y las que hacen esfuerzos verticales como Imagenio. Estas soluciones permiten una difusión multicast que hacen un uso más eficiente de los recursos de la red, pero por contra necesitan un STB. Por otra parte están los servicios "Over The Top" (OTT) como Youtube, que son muy ineficientes, pero sus requisitos de utilización son muy bajos.

En cuanto a la electrónica de consumo, su actual grado de desarrollo permite en la actualidad ejecutar aplicaciones sobre un sistema operativo (SO) común y abierto de forma que el STB puede desaparecer.

En los últimos años, aprovechando las múltiples ventajas y utilidades de la nube, han surgido sistemas de banda ancha que entregan video y audio sin necesidad de que un operador reserve recursos y en muchos casos sin necesidad de un hardware adicional. A estos servicios se puede acceder desde cualquier dispositivo que disponga de un sistema operativo o middleware. Por el contrario la mayoría de estos servicios sólo ofrecen videos a la carta, y además no tienen muy desarrollada la seguridad del contenido que distribuyen.

El objetivo del proyecto es realizar un estudio sobre una solución para sustituir los STBs actuales, necesarios para IPTV y utilizados para poder visualizar la televisión de pago, por un STB virtual alojado en la nube. Este STB evita tener múltiples dispositivos físicos, pero deberá mantener la seguridad de los STB tradicionales, como son el acceso condicional y la gestión digital de los derechos del contenido (DRM). Como parte del pfc se va a evaluar su viabilidad en Google TV, que es la plataforma Smart TV de Google. Esta plataforma cuenta con un SO Android, el cual es un entorno de desarrollo abierto y conocido que facilitará su implementación.

También se ha realizado una prueba de concepto que expusiera la funcionalidad del sistema. Para ello se han utilizado los lenguajes y interfaces más modernos con el fin de que sea lo más representativa posible. Finalmente se documentan los problemas que aparecieron debido al grado de madurez de algunas de las tecnologías empleadas para su implementación.

Abstract

Nowadays the services of television over IP are having a great accueil on the market. This is due to its easy implementation and the growth of the bandwidth that arrives to the home, so that it is possible to provide a fluid emission of the content. IPTV service providers advantage from their actual infrastructure to guarantee dedicated bandwidth for the television retransmission. These broadband service providers have integrated voice, information and television services by means of the IP network protocols. This integration was called "Triple Play ".

IPTV offers live streaming channel, time-shifted television and video on demand. It only requires a Set-Top-Box (STB) which is in charge of coding the signal and providing the necessary security. Regarding the strategy of service, there are several trends. On the one hand, there are solutions which use standard bodies such as DVB, OIPF or TISPAN and vertical platform, which is the case of Imagenio. These solutions allow multicast, which uses the network resources more efficiently, but they relay on the STB. On the other hand, there are "Over The Top" (OTT) services as Youtube, which are highly inefficient, but their usage requirements are very low.

Regarding the consumer electronics, its current grade of development allows currently run apps over a common and open operative system (OS), so that the STB may disappear.

In recent years, there are arisen broadband systems that deliver the video and audio without needing a service provider. Thanks to the multiple cloud advantages and spread usage, dedicated resources are guaranteed and in many cases without needing an additional hardware.

However the most of these services only offer videos on demand, and furthermore they do not provide the desired security for the content that they distribute.

The goal of this project is to make a report about a solution which may allow the replacement of current STBs, necessary for IPTV and for providing pay television, by virtual STBs hosted in the cloud.

The proposed STB prevents from having several physical devices while supporting the same security levels, conditional access and digital right management (DRM) as traditional STB does. As part of this thesis we will evaluate its viability on Google TV, which is Google's Smart TV platform.

This platform relays on Android OS, which is an open and acquaintance development environment, which is going to ease its implementation.

We have also carried out a prove of concept which consist on exemplifing the functionality of the system. For this purpose we have been using the latest languages and interfaces so that it was as reliable as possible. To conclude we present the confronted issues mainly due to the degree of maturity of some of the technologies used for his implementation.

Índice General

Índice de figuras

Índice de tablas

Capitulo 1

Introducción

1.1 Motivación del proyecto

En los últimos años, el crecimiento del ancho de banda del internet que llega a los hogares, principalmente debido a la implantación de fibra óptica y a la tecnología IP, ha abierto un abanico de posibilidades para los proveedores de televisión, ya que con una velocidad de bajada mayor, el usuario puede visualizar el contenido multimedia por internet sin cortes y con mejor calidad. Estos proveedores cada vez son más utilizados debido a la creciente demanda, por parte de los usuarios, de poder visualizar el contenido multimedia en cualquier momento y lugar, esto está provocando un cambio de prestaciones en el modelo de negocio de muchos fabricantes y proveedores de televisión a través de fibra óptica y por satélite, añadiendo la funcionalidad de "Catch-up TV" la cual posibilita ver un contenido concreto en los días posteriores a su emisión.

Otro hecho que ha acompañado paralelamente a este, es el aumento de dispositivos desde los cuales se pueden visualizar contenidos multimedia como son los Smartphones, Tablets y Smart TV, lo cual deriva en una amalgama de dispositivos y fabricantes , que soportan diferentes formatos. Sus pantallas tienen diferentes propiedades y tamaños, por este motivo la red necesita tener las prestaciones suficientes para poder disponer del formato correcto cuando el usuario requiera visualizar el contenido.

Contrariamente a esta diversidad de dispositivos, se ha experimentado un fenómeno de convergencia de usos, es decir, los usuarios quieren que un mismo dispositivo sirva para todo, un teléfono móvil a parte de llamar debe tener cámara de fotos, reproductor de mp3, conexión a internet, un sistema operativo, diversas aplicaciones sociales y de entretenimiento, y muchas otras que se han ido incorporando en los últimos años y que no cabe duda que van a seguir creciendo. De la misma forma, cada vez más, se requieren televisiones desde las que se puedan reproducir videos por USB, ver fotos, con acceso a internet y aplicaciones a cualquier otro servicio que se pueda usar en otro dispositivo.

En relación a los anterior, la evolución de la electrónica de consumo ha propiciado que las Smart TV tengan un precio competitivo de forma que pueda ser adquirida por la mayor parte

de la sociedad. Aquí es cuando entra en juego Google TV, que es la plataforma Smart TV de Google. Esta plataforma cuenta con un SO Android desde el cual se puede acceder a las mismas aplicaciones que desde un Smartphone ó Tablet. Además permite una fácil migración desde aplicaciones ya existentes en estos dispositivos.

Adicionalmente, cabe reseñar que el comercio electrónico y la venta por internet han realizado un crecimiento exponencial en los últimos años y las expectativas son que sigan creciendo en los años venideros. Los consumidores, seducidos por las ofertas de internet y por la gran comodidad de poder comprar a golpe de click, han espoleado las ventas en este sector. Todo parece indicar que el consumidor se está volviendo cómodo a la hora de comprar, lo cual beneficia a este sector, por lo que el acceso a internet de las Smart TV y la gran cuota de mercado de las televisiones, (prácticamente todos los hogares tienen una televisión en casa) hace que sea una plataforma idónea para vender productos.

La motivación de este proyecto es poder proporcionar al usuario lo que desea, cuando desea y a un coste reducido, que pague sólo por lo que ve. Debido a la complicada situación económica actual, muchos consumidores no pueden permitirse el coste de contratar televisión de pago con una tarifa plana, aunque sí podrían costearse ver un partido o ver una película en un instante determinado. Esto actualmente no es posible, pues para poder acceder a una canal de pago, se requiere haber contratado un servicio con un proveedor de IPTV, cable o satélite y tener un decodificador previamente instalado. Otra alternativa son las aplicaciones OTT, estas aplicaciones son muy ineficientes y carecen de seguridad alguna.

Sin embargo debido a la evolución de la electrónica de consumo, se permite en la actualidad ejecutar aplicaciones sobre un sistema operativo (SO) común y abierto de forma que no es necesario una soporte físico para poder acceder al contenido, pero adicionalmente necesitaremos algunos módulos de seguridad, de protección del contenido, acceso a prestaciones como la EPG, estos módulos se pueden externalizar, pudiendo agruparlos en una plataforma como la nube que les da soporte, de aquí en adelante a este concepto le llamaremos STB virtual.

1.2 Objetivos

El objetivo general de este proyecto es evaluar de una manera fiable la viabilidad de usar un STB virtual implementado en la nube como un sistema alternativo a los propuestos actualmente por los suministradores de canales de televisión de pago.

Además se hará un estudio de compatibilidad con las tecnologías idóneas y se propondrán alternativas si estas no fueran viables.

A continuación se enumeran una serie de objetivos que se persiguen en la realización del proyecto:

- Llevar a cabo un estudio de las principales herramientas que pueden servir tanto para la realización del STB virtual como sobre los formatos y lenguajes más adecuados a utilizar.
- Definir el concepto de STB virtual y los retos que supone para la tecnología existente.
- Realizar el diseño del STB virtual, se presentará un diseño modular y funcional de su arquitectura.
- Desarrollar una aplicación sobre un SO en la que se pueda visualizar contenido multimedia alojado en un servidor externo.
- Probar el correcto funcionamiento de la aplicación y explorar compatibilidades con canales en directo.
- Desarrollar una web empotrada en un browser desde el cual se pueda acceder al contenido multimedia.
- Probar su correcto funcionamiento así como sus posibles alternativas.
- Evaluar y desarrollar la compatibilidad de la aplicación con la web.
- Estudiar si Google TV es una herramienta idónea para la implementación de la aplicación.

1.3 Contenido de la memoria

La memoria está estructurada de la siguiente manera:

- **Capítulo 1** en el que nos encontramos, en el se justifica el porqué se va a realizar el proyecto y a que se pretende dar solución. También se establecen unos objetivos a cumplir durante su realización.
- **Capítulo 2** se introduce el estado actual de las diferentes tecnologías y servicios que se utilizan en este proyecto. Se especifican las diferentes soluciones actuales para la distribución del contenido en la nube así como las diferentes plataformas existentes. También se detalla cómo se realiza el tratamiento del contenido multimedia que se transporta a través de internet. A continuación se tratan los principales problemas que tienen las soluciones de la nube como son la privacidad y la seguridad.
  Finalmente se introducen algunos de los lenguajes y tecnologías utilizados.

- **Capítulo 3** se realiza una descripción sobre qué vamos a implementar, introduciremos y detallaremos el concepto de STB virtual, así como diferentes escenarios que se pueden implementar. Además avanzaremos una serie de retos con los que a priori nos vamos a encontrar a la hora de la realización del proyecto.
- **Capítulo 4** se realiza el diseño del STB virtual, para ello se describirá la arquitectura modular y funcional del sistema.
- **Capítulo 5** se lleva a cabo el estudio de compatibilidad de todo el sistema, así como los inconvenientes que ha tenido elegir un determinado lenguaje o formato y no otro. Se incluirán diversas tablas para facilitar su lectura y comprensión. Adicionalmente se realizará una prueba de concepto del sistema diseñado.
- **Capítulo 6** se documentan las pruebas realizadas. En este capítulo se recoge tanto si la prueba de concepto de que nuestro sistema es o no viable, como diversas pruebas adicionales que se han realizado.
- **Capítulo 7** se describen las diferentes fases que ha tenido el proyecto así como las tareas realizadas, los principales problemas que se han encontrado y los resultados obtenidos.
- **Capítulo 8** se establecen las conclusiones acerca del proyecto y se realiza un análisis de las posibilidades de investigación en un futuro.

Capítulo 2

Estado del arte

2.1 Introducción

La aparición del Cloud Computing ha cambiado la forma de entender internet. Actualmente un gran porcentaje de los usuario de internet usan estás soluciones. En un principio la mayoría de las soluciones estaban orientadas a su uso como un sistema de almacenamiento, pero actualmente se le está dotando de nuevos usos. Uno de los más demandados es compartir el contenido multimedia, de forma que el usuario lo pueda reproducir en otro lugar u otro dispositivo adaptando el contenido.

Aunque hay muchas ventajas para adoptar el Cloud Computing, también tiene algunos problemas en los que se están trabajando. Los problemas más importantes están relacionados con la seguridad del contenido que se aloja en ellos.

Por otra parte, la televisión está experimentando un cambio. Está intentando lograr la interactividad con los usuarios y quiere dejar de ser una vía de comunicación unidireccional, para convertirse en bidireccional, este fenómeno ha sido impulsado por las Smart TV y los proveedores de IPTV. Este hecho ha contribuido el desarrollo de múltiples soluciones para garantizar el acceso condicional y la DRM.

2.2 STB

Un STB es un dispositivo electrónico que se encarga de recibir una señal digital o analógica desde alguna de las diferentes redes como satélite (DVB-S), terrestre (DVB-T o DVB-H), móvil (MBMS) o por internet (IMS IPTV). Posteriormente para interconectar el STB con el CAM se utiliza el estándar DVB-CI. Los STB pueden tener varias ranuras CI para varios proveedores. Cada proveedor utilizará su propio CSA para enviar la información ofreciendo seguridad añadida.

Más adelante, entraremos en detalle de la seguridad que concierne al contenido que procesa un STB, pero en este apartado nos limitaremos a introducir cuál es la arquitectura básica de un STB.

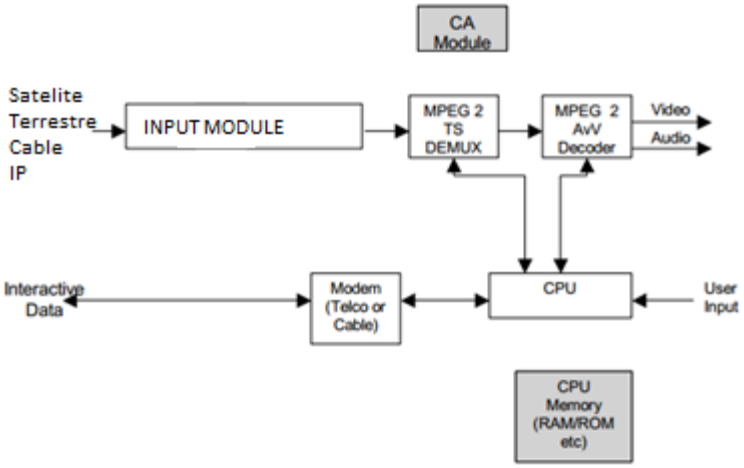

Figura 1: Arquitectura modular de un STB [Sundareshan09]

El funcionamiento de la arquitectura es el siguiente. El STB recibe la señal de video proveniente en cualquiera de los estándares establecidos. Posteriormente pasa por el demultiplexor y decodificador del MPEG2-TS, a la salida del cual tendremos el video y el audio para ser enviado a la pantalla.

Adicionalmente, los STB suelen disponer de algún tipo de modem. Este permite enviar y recibir datos en caso de que el video recibido fuera broadcast y no dispusiera canal de retorno, por ejemplo TDT. En el caso de la IPTV al ser multicast cuenta con un canal de retorno, por lo tanto no es necesario este módulo.

Por último la CPU se encarga de manipular los datos y de realizar las operaciones necesarias para la reproducción y distribución del contenido. Esta CPU utiliza un SO por encima de la capa hardware.

Esta arquitectura modular, con la ayuda del modulo CA que vemos en la parte superior de la figura 1, nos permitirá proporcionarle el control de acceso requerido por parte de este tipo de aplicaciones.

Además de la arquitectura modular, en este apartado, también vamos a analizar cuál es la arquitectura funcional del sistema básico de un STB.

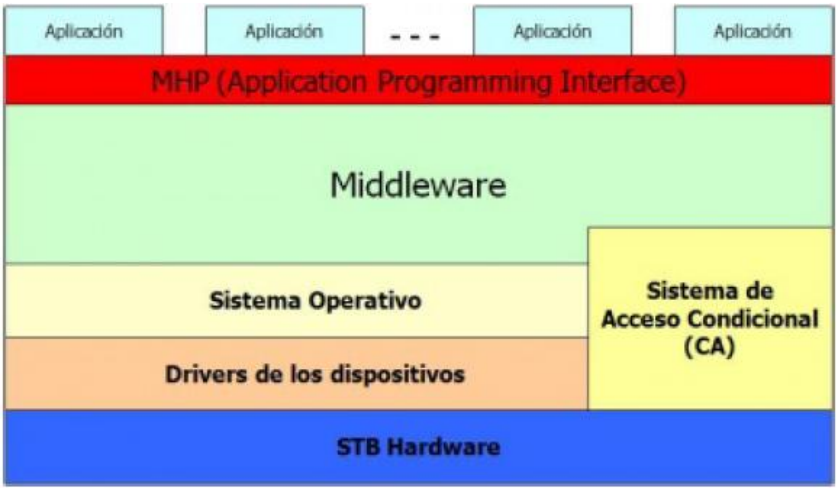

Figura 2: Arquitectura funcional de un STB [Aponte09]

Como se observa en la figura 2, el STB consta de varios niveles:

- **Nivel de Hardware**: Está compuesto por todos los componentes físicos del STB: la CPU, el demodulador, la memoria y el modulo de acceso condicional (CAM), entre otros.
- **Nivel de Sistema Operativo**: este se nivel se comunica con el hardware, para que el SO sea capaz de demultiplexar las tramas MPEG se necesita un SO en tiempo real.
- **Nivel de Middleware o Plataforma**: este nivel se encarga de comunicar al SO con las aplicaciones del abonado. Esta capa reduce la complejidad del desarrollo de aplicaciones, pudiéndose emplear un API común, aislando la capa de aplicación de los detalles de los componentes de hardware y otras redes subyacentes.
- **Nivel de aplicación**: en este nivel se encuentran las aplicaciones instaladas por el usuario. Esta capa sólo se utilizará cuando se quiera acceder a una de ellas, el resto del tiempo permanecerá idle.

2.3 Smart TV

El concepto de Smart TV es el de dotar a un dispositivo pasivo como era la televisión, de una conectividad bidireccional que le proporcione interactividad, en cierto modo, que la haga "inteligente".

Básicamente se considera Smart TV a una televisión que tiene conectividad a internet y un sistema operativo desde el cual se pueden instalar aplicaciones de usuario.

Estas soluciones trasladan el contenido de la web a la televisión, tales como Google TV [Goo11], Apple TV [App11], Roku [Roku10] y Boxee Box [Box11] las cuales combinan el acceso a canales en directo con una amplia selección de contenidos guardados que se pueden

visualizar. Gracias a esto el usuario tiene a su disposición una gran cantidad de programas, que puede alternar, creando y manejando distintas listas de reproducción. En la figura siguiente se puede ver una comparativa de estas soluciones.

Figura 3: Comparativa de STBs [lhbeststb]

Estas soluciones permiten añadir servicios para conectar la televisión con las redes sociales (p.ej. Twitter) galerías de fotos (p.ej. Flicker), etc. Las tecnologías Smart TV están siendo desarrolladas por multinacionales como Sony [Son11], Panasonic [Pan11], LG [Lg11] o Samsung [Sam11]. Estas soluciones incluyen un nivel web mejorado, que permite conectar la web con las aplicaciones que han sido específicamente realizadas para que la TV muestre a los usuarios su contenido digital elegido. Sin embargo, la mayoría de esas soluciones, pertenecientes a compañías de fuera de la UE, son independientes y no son compatibles, lo cual deja a decisión del usuario cómo tiene que organizar y distribuir los derecho de visionado y uso del contenido.

Actualmente, las compañías desarrolladoras de televisiones, están apostando por convertir la televisión en Smart TV sin necesidad de un STB. Esto se realiza incorporando a las televisiones un microprocesador y la arquitectura suficiente para poder soportar un sistema operativo básico. Al principio cada compañía diseñaba su propio sistema operativo propietario, normalmente bajo un kernel Linux, lo que provocó que el volumen de aplicaciones existentes fuese muy reducido con respecto al volumen de aplicaciones para dispositivos móviles. Samsung fue la primera que apostó por incluir Google TV como sistema operativo dentro de su televisión. Actualmente LG y Sony ya han empezado a sacar modelos con el mismo sistema operativo.

Esta decisión de tres de las grandes compañías productoras de televisiones de adoptar Google TV como sistema operativo así como la facilidad para migrar aplicaciones de plataformas móviles a la televisión, puede proporcionar el empujón definitivo al proyecto de

Google TV que de momento no acaba de despegar. A continuación vamos a enumerar una serie de motivos por los cuales Google TV no se ha implantado con fuerza en los mercados:

- Las compañías productoras de televisión empezaron a fabricar Smart TV con otros S.O. lo que propició que esos clientes no se compraran un STB.
- Introducción en el mercado de USB Android que no son Google TV. Esto significa televisiones con un sistema Android distinto al 3.1 y 3.2, para los cuales está desarrollado el Add-on de Google TV.
- Google inicialmente sólo adaptó Android 3.1 a la TV incluyendo el Google TV-Addon, en 2013 Google adaptó también Android 3.2, sin embargo todavía está muy lejos de su última versión, por eso muchas compañías decidieron introducir una versión superior como SO de sus televisiones, que aunque no esté estrictamente optimizado para estos dispositivos ofrece un buen rendimiento.
- Dificultad de utilización de los nuevos mandos a distancia. Muchos clientes están insatisfechos por la dificultad para utilizarlos. Por ello, ya se están fabricando algunos modelos más sencillos y prácticos.
- El precio es el principal problema. Los STB son muy caros actualmente, y la población prefiere invertir más en la calidad de la televisión que en su SO.

Todos estos problemas, se han tenido que solventar, y algunos todavía se están solventando. Sin embargo lo que parece claro es la tendencia a adoptar Android como SO de las televisiones.

2.4 Servicios sobre la nube

En estos últimos años se han sucedido una gran variedad de soluciones de Cloud Computing con diferentes propósitos y objetivos, con el fin de solventar algunos de los aspectos relacionados con la gestión y distribución del contenido. Algunos de los cuales son iCloud [ICl11], Dropbox [Dro11], Pogoplug [Pog11], Tonido [Ton11], Zumo Drive [Zum11]. Estas soluciones son principalmente de almacenamiento pero han conseguido proporcionar al usuario una gestión y distribución del contenido adecuada a sus necesidades.

Por otro lado, la rápida evolución de la electrónica de consumo ha propiciado un incremento del número de dispositivos que pueden acceder al contenido multimedia. Por este motivo, hay que garantizar que el contenido es adaptable a cualquier tipo de dispositivo existente en el mercado actual. Una de las soluciones para adaptar el contenido es el

transcoding, que crea una copia de un video en diferentes formatos compatibles con la mayoría de los dispositivos, facilitando así su accesibilidad.

Para dar solución a esta necesidad Cloud Computing proporciona varias soluciones altamente escalables, como aplicaciones OTT o algunas plataformas de contenidos multimedia en la nube. Estas soluciones se explicarán en los dos apartados posteriores.

2.4.1 Contenido Over-The-Top

El contenido OTT es un sistema para poder hacer llegar al usuario final video y audio sin necesidad de que un operador se haga cargo de su control y distribución. El proveedor del contenido OTT, sabe el contenido de los paquetes IP, pero no es capaz de controlar la capacidad de visión, los derechos de autor y la redistribución del contenido, las cuales son el objetivo de los proveedores de IPTV, DVB-S, DVB-C y DVB-T. OTT se refiere al contenido que llega desde un tercero, tales como NowTV [NowTV12], Netflix [Netflix11], WhereverTV [Wherever11] o Hulu [Hulu09] y es entregado al dispositivo del usuario final. La responsabilidad de entregar los paquetes IP recae sobre el operador.

Por estos motivos las aplicaciones OTT no garantizan la entrega del contenido, ya que el estado de la red y la reserva de recursos son transparentes a este tipo de aplicaciones. Además son muy ineficientes debido a que realizan multitud de solicitudes HTTP y además carecen de cualquier tipo de seguridad a la hora de enviar su contenido.

Los consumidores pueden acceder al contenido OTT a través de dispositivos con conexión a internet tales como ordenadores, portátil, Tablets, Smartphones, STBs y videoconsolas.

2.4.2 Plataformas de almacenamiento en la nube

Hay una serie de plataformas de almacenamiento de contenido multimedia en la nube como Dailymotion Cloud [DmCloud], Stream Nation [StrmNat08], Vzaar [Vzaar08], Viddler [Viddler08] e inXtron [Inx11]. Estas soluciones no se limitan a almacenar el contenido multimedia, cuando el usuario sube un video, lo transcodifican, haciendo que su video sea compatible con múltiples dispositivos e incluso lo codifica para alta definición.

Stream Nation permite al usuario poder almacenar enlaces de servicios OTT para poder reproducir su contenido como son Youtube, Vimeo o Daily Motion. Además hace una copia de tus archivos en Amazon para garantizar la seguridad de los mismos.

Viddler proporciona un núcleo de funcionalidades comunes a todos los usuarios y luego permite ir incorporando complementos como poner publicidad, filtrar el contenido por la

localización del usuario o bloquear IPs entre otras. Todos estos complementos ayudan a mejorar la experiencia del usuario.

Vzaar añade un nivel extra de seguridad llamado RTMPe. Este nivel encripta el streaming y además hace una verificación SWF, proporcionando al video una seguridad añadida.

inXtron propone la idea de un servidor personal en la nube, incluyendo un servidor multimedia UPnP-AV para reproducir archivos multimedia a través de red local, capacidad para crear listas de reproducción y posibilidad de acceder desde cualquier lugar al servidor a través de un portal web. Además, los usuarios pueden crear distintas carpetas para compartir los ficheros con su familia y amigos. inXtron hace énfasis en que al ser un servidor propietario y no estar alojada la información en un servidor externo será más seguro.

Finalmente una reciente propuesta [Diaz11] proporciona un middleware que puede ser instanciado en un STB o un Gateway local, llamado Media Cloud, para clasificar, buscar y compartir contenido multimedia a través de su dominio local en la nube. Sus objetivos son proporcionar la máxima interoperabilidad con el mínimo esfuerzo del usuario, permitir acceder de forma transparente desde diferentes dispositivos a la red local y comunicarse como si ellos estuvieran en esa misma red local.

2.5 Tratamiento del contenido multimedia

Los servicios multimedia para dispositivos móviles se han disparado en la última década. Los usuarios actualmente quieren poder reproducir el contenido multimedia en cualquier sitio y en cualquier dispositivo desde el que estén conectados, lo cual ha producido un crecimiento de los servicios multimedia y aplicaciones que tratan de dar solución a esta demanda.

Además cada vez más contenidos se crean por los usuarios de sitios web, liderando un crecimiento de nuevos modelos de negocio y de maneras de compartir el contenido.

Este progreso ha sido posible gracias a unas eficientes técnicas de compresión de imágenes y videos.

2.5.1 Streaming Media

El nuevo estándar MPEG-DASH [Dashiso12] es la primera solución de Streaming basada en el protocolo HTTP que consigue unas tasas de transferencia adaptativas. Esto permite que diversos dispositivos como TV, Smartphones y Tablets puedan reproducir contenido multimedia de una misma web a una tasa de transferencia variable.

El MPEG-DASH se convertirá en un formato estándar para VoD ya que todos los grandes reproductores de contenido como Apple, Microsof, Netflix y muchas otras han participado en su desarrollo, lo cual hace prever que sustituirá a soluciones propietarias como HLS de Apple, Smooth Streaming de Microsoft, o HDS de Adobe.

En el mercado actual, aún no se usa MPEG-DASH. Por otro lado una de las soluciones que más popularidad ha alcanzado es HLS. Esta solución, con la cual se trabaja también en este proyecto, es una solución propietaria de Apple que permite enviar audio y video sobre HTTP de un servidor web a cualquier dispositivo Apple. Su arquitectura se muestra en la siguiente figura:

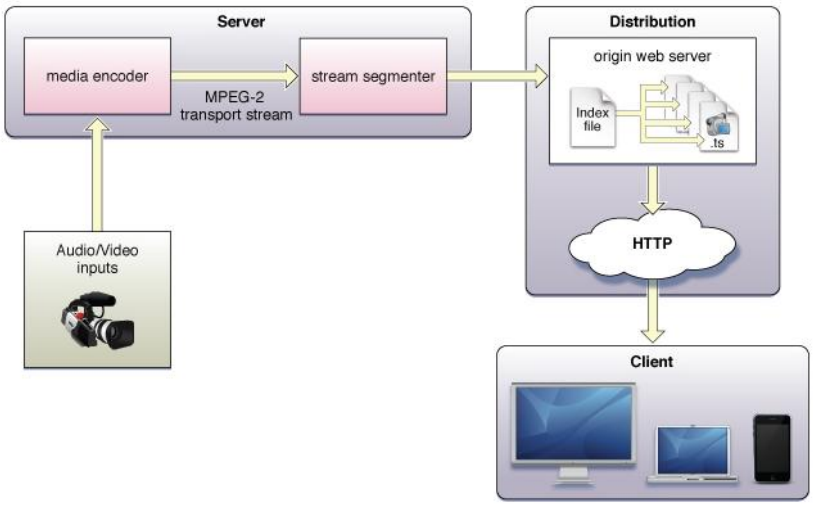

Figura 4: Arquitectura HLS

Como se puede apreciar en la figura, HLS posee una arquitectura muy sencilla. Este hecho unido a la popularidad del Iphone, Ipad y Macbook, han sido decisivos para la adopción de este estándar por una gran cantidad de usuarios.

Http Live Streaming soporta emisiones en directo y contenido pregrabado (VoD). Además también soporta múltiples streams a diferentes tasas de bit. Por último, también proporciona encriptación para el contenido multimedia y autenticación de usuario sobre HTTPS.

2.5.2 Compresión de video

HEVC, es un estándar de compresión de video que sucede a H.264/MPEG-4 AVC (Advanced Video Coding). Fue desarrollado conjuntamente por la ISO/IEC Moving Picture Experts Group (MPEG) e ITU-T Video Coding Experts Group (VCEG).

H.264/MPEG-4 AVC hizo posible que el video digital abarcara casi cada área que no fue cubierta previamente por el video H.262/MPEG-2, y que ha ido desplazando los otros estándares antiguos dentro del existente campo de aplicación.

Este estándar es muy utilizado para muchas aplicaciones, como:

- Emisión broadcast o señales de TV de alta definición (HD) por satélite, cable y sistemas de transmisión terrestre.
- Adquisición de video.
- Cámaras y aplicaciones de seguridad.
- Video a través de red móvil e internet.
- Disco blu-ray.
- Aplicaciones de conversación en tiempo real tales como video chats y videoconferencias.

Actualmente el estándar también cuenta con extensiones para video 3D, que será muy útil para futuras aplicaciones.

Sin embargo, debido a la gran popularidad que ha adquirido el video HD en los últimos años han ido surgiendo diversos servicios y formatos (p.ej. resoluciones 4k×2k o 8k×4k) que demandan una eficiencia de codificación mayor a la que H.264/MPEG-4 AVC puede ofrecer.

Además, el tráfico causado por las aplicaciones de video en dispositivos móviles, tabletas y PCs, así como la necesidad de transmisión para servicios VoD están creando dificultades a las redes actuales. El incremento de calidad y resolución del contenido multimedia también es un requisito de las nuevas aplicaciones móviles, que ven como los usuarios demandan cada vez mejor calidad.

HEVC ha sido diseñado esencialmente para cubrir todas las aplicaciones existentes de H.264/MPEG-4 AVC y centrarse en resolver dos problemas claves:

- El incremento de la resolución del video.
- El incremento de arquitecturas de procesamiento paralelo en varios núcleos.

La sintaxis de HEVC es genérica y debe ser generalmente adecuada para otras aplicaciones que no fueron mencionadas específicamente arriba [Sullivan12].

2.5.3 Codificación del contenido

A la hora de codificar los streams multimedia hay que tener en cuenta no sólo la gran diversidad de decodificadores y reproductores que hay en el mercado, sino también los múltiples dispositivos tales como PCs, Portátiles, Smartphone, Smart TV, Tablets y PDAs. Toda

esta amplia gama de dispositivos y reproductores así como sus características han de tenerse en cuenta a la hora de enviar el contenido multimedia al dispositivo final, de tal modo que el sistema de comunicación multimedia debe ser capaz de adaptar sobre la marcha el stream multimedia en función de las características y restricciones del receptor. Este tipo de servicios de comunicación multimedia son imprescindibles para el desarrollo eficiente de las aplicaciones que consumen contenido multimedia.

Para realizar esta adaptación sobre la marcha del contenido, se han implementado muchos algoritmos, pero una de las soluciones más sencillas es la codificación escalable del contenido multimedia. La codificación escalable se basa en crear unos niveles de representación del contenido. Existe un primer nivel que será el mínimo que se puede enviar que contiene la peor calidad, a partir de ahí podemos ir añadiendo niveles según la demanda del dispositivo y la tasa de transferencia que nos ofrece la red. Estos niveles adicionales contienen una mejora de la calidad (SNR o calidad escalable), resolución temporal (resolución escalable) o tasa de trama (tasa escalable).

Esta solución nos permite una perfecta adaptación a los diferentes escenarios que pueden ir surgiendo, permitiendo enviar la calidad óptima en función de las características del dispositivo y el estado de la red.

En la figura siguiente podemos ver gráficamente cómo funciona la codificación escalable.

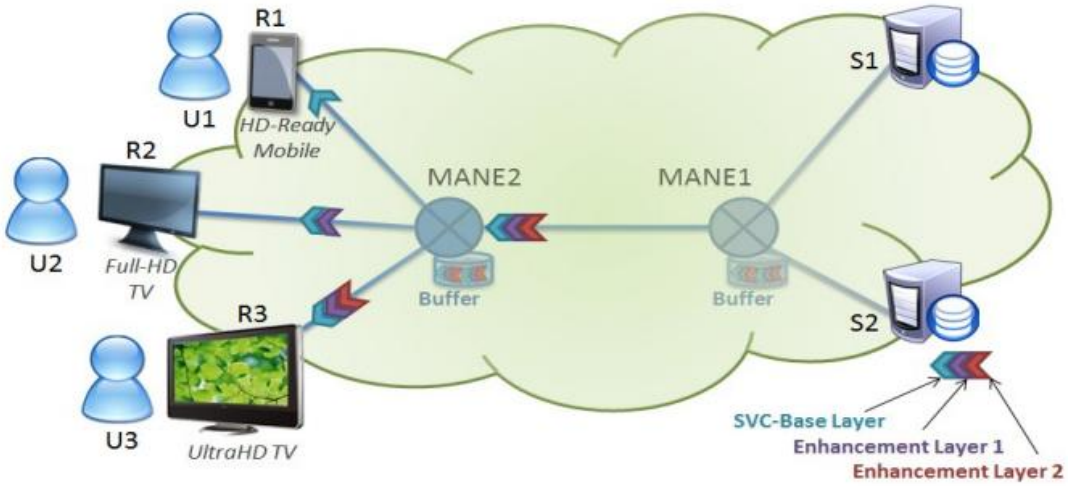

Figura 5: Codificación escalable [Graft13]

Hasta el 2007 que se estandarizó SVC no se utilizó este tipo de codificación, a pesar de que los estándares MPEG-2 y MPEG-4 ya se podían usar como extensión de H.264/AVC [Schwarz07]. Con H.264/SVC se comprobó que creando un stream escalable con la misma tasa de transferencia que un solo nivel de H.264/AVC, la pérdida de calidad era mínima.

A pesar de todas las ventajas que ofrece la codificación escalable, en la actualidad para transmitir la mayoría de streams de video se crea un solo nivel (usando p.ej. H.264/AVC). Esto es debido a que el cambio de codificación a H.264/SVC o HEVC SVC no es trivial y requiere una alta capacidad computacional. Por este problema la codificación en un solo nivel seguirá predominando en los próximos años, se seguirá utilizando el H.264/AVC.

2.5.4 Transcodificación

El proceso de adaptar el contenido multimedia en función de los dispositivos y del tipo y velocidad de la red, sin perder calidad, se denomina transcodificación [Vetro07].

En muchas ocasiones este proceso se limita a hacer una réplica del video que el usuario sube a la nube, en los diferentes formatos existentes, para que cuando se intente acceder al contenido desde otro dispositivo, se muestre inmediatamente y con una calidad óptima.

Para hacer frente a la variedad de configuraciones, condiciones y compatibilidades se aplican distintas operaciones de transcodificación:

- Transrating
- Un solo nivel sobrescribiendo el SNR de un stream escalable
- Transcodificación heterogénea
- Reducción de la resolución espacial de la trancodificación
- Reducción de la resolución temporal de la trancodificación

2.6 Internet Protocol Television (IPTV)

En la actualidad, los servicios de televisión sobre IP están encontrando una gran aceptación en el mercado.

La IPTV permite reproducir canales de televisión en directo, canales de televisión en diferido y video a la carta. Lo único necesario es un Set-Top-Box que decodificará la señal y proporcionará la seguridad necesaria.

A diferencia de los servicios OTT ya mencionado anteriormente, los cuales tienen grandes problemas de seguridad. A este protocolo sí se le puede proporcionar la seguridad requerida para este tipo de servicios.

Adicionalmente, dentro de los proveedores IPTV hay diversas soluciones. Por una parte están las que utilizan estándares, como:

- DVB

- OIPF
- TISPAN/ETSI

Por otra parte, soluciones que hacen esfuerzos verticales como son: Imagenio, Orange TV o ONO, entre otras.

2.6.1 Seguridad en IPTV

Además de la seguridad en la nube, también tenemos que tener en cuenta otros tipos de amenazas que pueden surgir cuando el contenido llega a los hogares para ser visualizado. Así como garantizar que la posterior gestión del contenido se realiza de forma correcta. Para ello el grupo DVB ha desarrollado dos estándares para garantizar la seguridad en la adquisición del contenido y en su trato posterior.

- **Protección en la adquisición**

Para proteger los datos de accesos no autorizados se emplea el estándar DVB-CA. Este estándar permite controlar el acceso de los subscriptores a los servicio de pago que tengan contratados.

Los dispositivos DVB-CA necesitan un demultiplexor MPEG-2 y hardware de acceso condicional. Las televisiones requieren una pantalla integrada y los STBs un modulo de exportaciones que envíe el contenido a la pantalla integrada.

En la figura siguiente podemos ver un esquema del funcionamiento de un dispositivo DVB-CA:

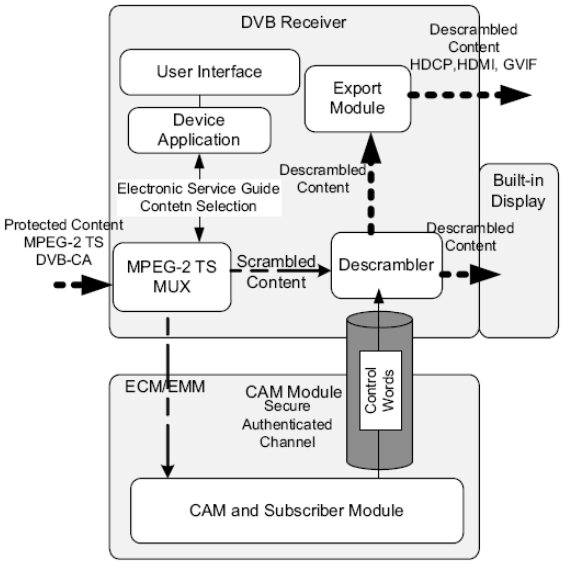

Figura 6: Arquitectura Modular de un dispositivo DVB-CA

Debido a la protección que DVB-CA realiza, se limita la flexibilidad del dispositivo, ya que cada dispositivo debe tener su propio módulo de subscripción. Para sobreponerse a esta limitación [Diaz09] propone un servicio DLNA que distribuye los mensajes de protección y autoriza mostrar el contenido en otros dispositivos.

Proteger al proceso de adquisición no es suficiente para garantizar el ciclo de vida del contenido, también debemos realizar una gestión digital de los derechos (DRM).

- **Protección después de la adquisición**

Una vez ha terminado el proceso de adquisición, el contenido debe ser manejado por DRM y los servicios de protección de copia deben evitar su distribución cuando esta no esté autorizada.

El estándar DVB-CPCM soporta el uso de reglas para definir:

- Dónde puede ser copiado o movido el contenido.
- Qué hardware es requerido.
- Cómo codificar localmente el contenido durante su transmisión.

Además DVB-CPCM proporciona interoperabilidad entre dispositivos DRM.

Dispositivos CPCM dentro de una familia, pueden establecer un Domino autorizado (AD) que limita la protección del contenido, gracias a esto se puede mover y copiar el contenido dentro de este dominio. CPCM también puede controlar en cuantos dispositivos se está reproduciendo el contenido simultáneamente. En la siguiente figura se representa un diagrama conceptual de un sistema CPCM.

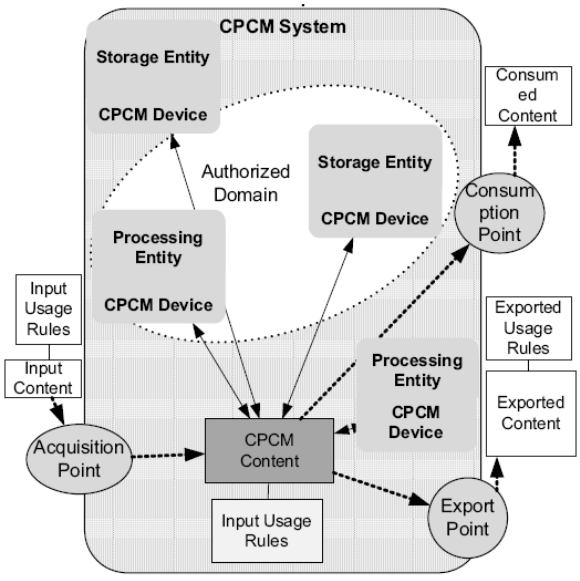

Figura 7: Diagrama de conceptual de un sistema CPCM

El contenido digital entra al sistema CPCM a través del punto de adquisición y se convierte en contenido CPCM. Dentro de este sistema el contenido puede ser copiado, movido, almacenado y procesado de acuerdo a las reglas de uso.

CPCM define cinco entidades funcionales, las cuales se pueden diferenciar en la figura anterior, que son:

- Punto de adquisición.
- Entidad de procesamientos.
- Entidad de almacenamiento.
- Punto de consumición del contenido.
- Punto de exportación.

En cuanto a la funcionalidad el sistema CPCM se divide en:

- Control de la seguridad.
- Manejo del contenido.
- Gestión del dominio autorizado.

Para poder distribuir el contenido en el dominio autorizado [Diaz10] propone una extensión de DLNA para usar DVB-CPCM, lo que garantiza la protección anteriormente explicada.

Por otro lado, estas soluciones que hemos analizado, solo permiten acceder al contenido en el dispositivo que tiene subscripción. Para solventar [Diaz09] propone un sistema local de gestión de las claves (HKMS). Este sistema permite compartir uno o varios CAMs con otros dispositivos, siempre y cuando tengan un descrambler.

La parte más vulnerable a ataques es la comunicación entre el CAM y el descrambler, para protegerlo se utiliza un canal de seguridad autenticado (SaC) por el cual viajará segura la clave.

Para proteger el audio y el video, se pasa mezcla con una palabra de control (CW) de 8 bits. Esta palabra de control se codifica con una clave de servicio (SK) dando lugar al ECM. Estos son enviados junto al audio y video mezclado.

Los proveedores usan EMMs que contienen el SK e información DRM codificada con una clave de tarjeta (CK). Mientras que los ECM son comunes para todos los subscriptores los EMM son específicos para un usuario.

Como se puede ver en la siguiente figura el audio, video y una referencia temporal se encapsulan en MPEG-2 TS.

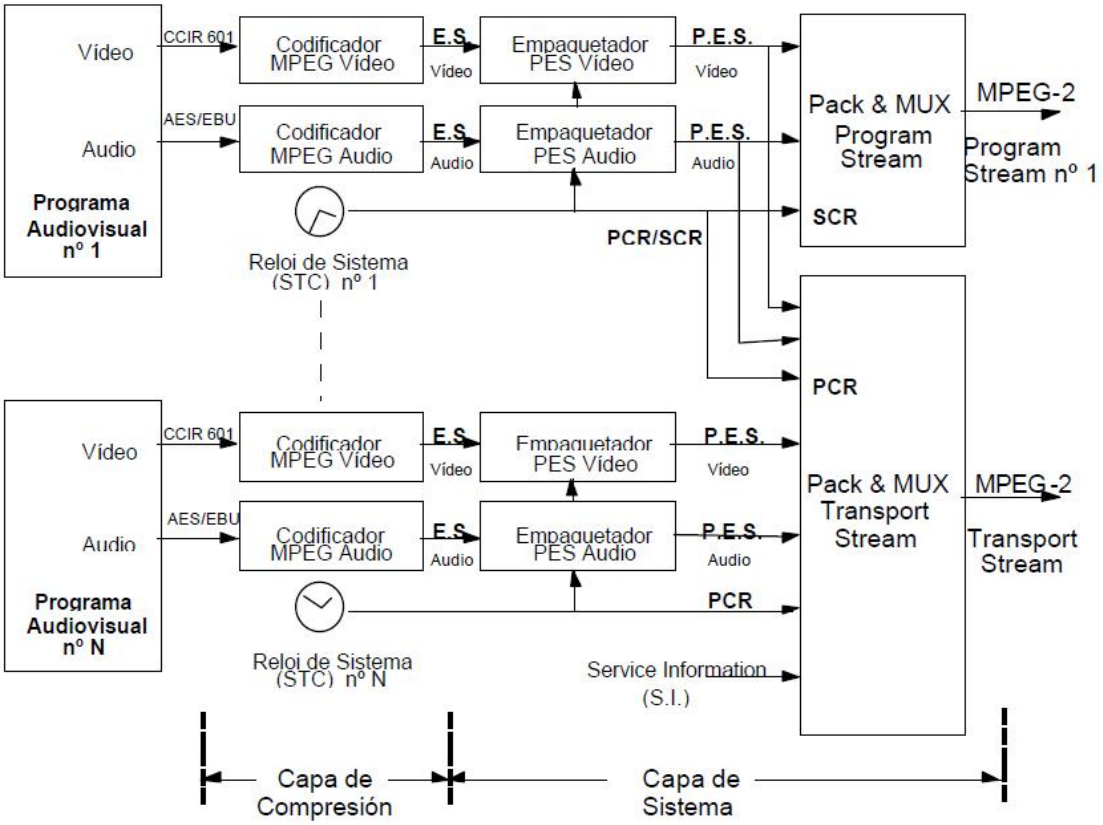

Figura 8: Encapsulación MPEG-2 TS [Mpeg2Wikitel]

MPEG-2 define un programa como un conjunto de Packetized Elementary Streams (PES).

Ya en el receptor, el CAM procesa los mensajes ECM y EMM, además son necesarios los detalles del subscritor (SC).

Durante la fase de establecimiento el CAM negocia un Sac con el descrambler. Con el canal ya seguro, el CAM envía la CW al descrambler.

Cuando la señal llega al STB se demultiplexa la trama MPEG-2, tras esto el descrambler restaura el video y el audio el cual es enviado al dispositivo para su visualización.

2.7 Lenguajes utilizados

En el desarrollo de este proyecto se han empleado diversos lenguajes de programación, en especial aquellos orientados al desarrollo de servicios web. A lo largo de esta sección se expondrán los aspectos principales y las características de dichos lenguajes.

2.7.1 HTML5

HTML5 (HyperText Markup Language, versión 5) es la quinta revisión del lenguaje de programación "básico" de la World Wide Web, el HTML. Esta nueva versión pretende remplazar al actual (X)HTML, corrigiendo problemas con los que los desarrolladores web se encuentran, así como rediseñar el código actualizándolo a nuevas necesidades que demanda la web de hoy en día. Alguna de sus ventajas más destacadas son:

- **Mejor visibilidad en contenidos móviles**.
- **Estructura del cuerpo**: La mayoría de las webs tienen un formato común, formado por elementos como cabecera, pie, navegadores, etc. HTML 5 permite agrupar todas estas partes de una web en nuevas etiquetas que representarán cada uno de las partes típicas de una página.
- **Bases de datos locales**: el navegador permitirá el uso de una base de datos local, con la que se podrá trabajar en una página web por medio del cliente y a través de un API. Es algo así como las Cookies, pero pensadas para almacenar grandes cantidades de información, lo que permitirá la creación de aplicaciones web que funcionen sin necesidad de estar conectados a Internet.
- **Web Sockets**: Proporciona una forma mucho más rápida de transferencia de datos en línea y comunicación, pudiendo recibir actualizaciones en tiempo real desde servidores.
- **Mejor acceso sin conexión**: Con aplicaciones ricas de Internet que almacenarán más información, como mensajes de correo electrónico para ver incluso sin conexión.
- **Ubicación geográfica**: posibilidad de ubicar al usuario que navega el sitio.
- **Sin plugins**: Con HTML5 además es posible reproducir de forma nativa en el navegador audio y video.
- **Fin de las etiquetas de presentación**: todas las etiquetas que tienen que ver con la presentación del documento, es decir, que modifican estilos de la página, serán eliminadas. La responsabilidad de definir el aspecto de una web correrá a cargo únicamente de CSS.

2.7.2 CSS3

CSS es un lenguaje para definir el estilo o la apariencia de las páginas web, escritas con HTML o de los documentos XML. CSS se creó para separar el contenido de la forma, a la vez que permite a los diseñadores mantener un control mucho más preciso sobre la apariencia de las páginas. A partir del año 2005 se comenzó a definir el sucesor de esta versión, al cual se lo

conoce como CSS3 o Cascading Style Sheets Level 3. Actualmente en definición, esta versión nos ofrece una gran variedad de opciones muy importantes para las necesidades del diseño web actual. Desde opciones de sombreado y redondeado, hasta funciones avanzadas de movimiento y transformación. CSS3 es el estándar que dominará la web por los siguientes años. Algunas de sus ventajas son:

- **Efectos Visuales**: Elementos como esquinas redondeadas, sombras y demás, son mucho más fácil de codificar.
- **Mejores Estilos para elementos**: La distribución y el aspecto de los elementos se mejora enormemente.
- **Cajas con sombra**: La propiedad de CSS3 box-shadow permite agregar un efecto de sombra sin usar imágenes a un elemento seleccionado. Box-shadow es soportado actualmente del Safari 3+ y Firefox 3.1+.
- **Múltiples columnas**: Este módulo de CSS3 permite colocar los textos en varias columnas de forma mucho más simple usando las propiedades: -moz-column-count and -moz-column-width.

2.7.3 JSF

El objetivo de la tecnología JavaServer Faces es desarrollar aplicaciones web de forma similar a como se construyen aplicaciones locales con Java Swing, AWT (Abstract windowToolkit), SWT (Standard Widget Toolkit) o cualquier otra API similar.

La tecnología JavaServer Faces constituye un marco de trabajo (framework) de interfaces de usuario del lado de servidor para aplicaciones web basadas en tecnología Java y en el patrón MVC (Modelo Vista Controlador).Los principales componentes de la tecnología JavaServer Faces son:

- **Similitud con HTML**: El código JSF con el que creamos las vistas (las etiquetas jsp) es muy similar al HTML. Lo pueden utilizar fácilmente desarrolladores y diseñadores web.
- **Compatibilidad con JSP**: JSF se integra dentro de la página JSP y se encarga de la recogida y generación de los valores de los elementos de la página.
- **Mejora las operaciones internas**: JSF resuelve validaciones, conversiones, mensajes de error e internacionalización.
- **Integración de javascript**: JSF permite introducir javascript en la página, para acelerar la respuesta de la interfaz en el cliente (navegador del usuario).

- **Tecnología extensible**: Se pueden desarrollar nuevos componentes a medida, también se puede modificar el comportamiento del framework mediante APIs que controlan su funcionamiento.

Capítulo 3

Conceptos

3.1 Introducción

En este capítulo se presenta una descripción general del sistema que se ha desarrollado en el proyecto. En él se va a describir qué se ha implementado y el por qué se ha decidido hacer de esa manera, justificando así la implementación.

En este apartado también se van a enumerar una serie de retos que a priori se van a encontrar en la realización del sistema de la forma descrita. En apartados posteriores se hablará acerca de cómo se han podido solventar estas dificultades.

3.2 STB virtual

Aunque la tecnología en las televisiones ha avanzado mucho, actualmente seguimos necesitando un decodificador externo para poder acceder a la televisión de pago. Ya que aunque se han implementado muchas alternativas en software que han aumentado la seguridad de la red, la forma más segura para proporcionar el acceso condicional y garantizar la gestión digital del contenido sigue siendo mediante hardware.

Este proyecto desarrolla la idea de substituir el decodificador externo por un STB virtual, el cual está alojado en la nube. Pero este STB debe seguir teniendo todas las funcionalidades del STB físico.

El creciente asentamiento en la sociedad de tendencias como “pay as you go” y la televisión a la carta, están produciendo un cambio en la venta o distribución de este tipo de contenido. Para intentar llegar a más clientes, se propone una solución abierta y altamente escalable, la cual permite a un usuario desde su televisión poder acceder a un canal concreto para ver una película o partido que se está emitiendo en un instante determinado o en diferido, independientemente del proveedor del contenido, y sin necesidad de tener contratado previamente un servicio adicional que le facilite el decodificador necesario para poder visualizar el contenido.

En los últimos meses se han desarrollado multitud de soluciones de integración vertical y OTT, algunas de las cuales son similares. Estas soluciones también mezclan como en este

proyecto, televisión en directo, con películas o capítulos de series ya emitidos, para poder reproducirlos cuando el usuario lo desee. Pero estas soluciones no satisfacen el problema al cual da solución este proyecto, por dos motivos principales:

- **Falta de escalabilidad**: estas soluciones verticales son propietarias, lo cual implica que si hay 5 proveedores de IP el usuario tendría que tener 5 aplicaciones instaladas y tener una lista de qué canales ofrece cada una. En estas soluciones el contenido no se presenta de una forma global, por lo tanto, no es posible ver el contenido de todos los canales de una sola vez, ya que muchos están emitidos por diferentes proveedores.
- Estas soluciones necesitan ser **contratadas previamente** para poder acceder a él. Este proyecto propone una solución abierta en la que se pueda ver el contenido que ofrece gratuitamente y dejar que el usuario pueda pagar estrictamente por lo que ve.

Adicionalmente hay que crear un interfaz de usuario, para que el contenido también pueda ser visualizado desde cualquier dispositivo electrónico que tenga una conexión a internet y un navegador. Para aumentar la seguridad y mejorar la integración del contenido al dispositivo se creará una aplicación para Smart TV. Las televisiones con SO integrado están experimentando un gran aumento. Esto se debe a que el mundo de las aplicaciones OTT está en pleno auge, el usuario utiliza aplicaciones para todo, además se familiariza más con ellas y le es más cómodo acceder a los contenidos. Es por eso que se crea una aplicación que enlazará con el browser empotrado para mostrar su contenido.

A continuación vamos a describir varios escenarios que muestran algunas de las ventajas que se podrían obtener utilizando el STB virtual. En cada escenario se comentarán las ventajas e inconvenientes que tiene cada uno.

3.2.1 Escenario básico

El escenario básico representa la idea general, de cuál es el funcionamiento del STB virtual.

Uno de los objetivos en los que hemos hecho hincapié a lo largo de la memoria es la necesidad de que sea un servicio al que se pueda acceder desde múltiples dispositivos. Actualmente, el usuario puede usar cualquiera de sus dispositivos para acceder al contenido, y para ello se ha debido implementar previamente las opciones necesarias para su correcta utilización.

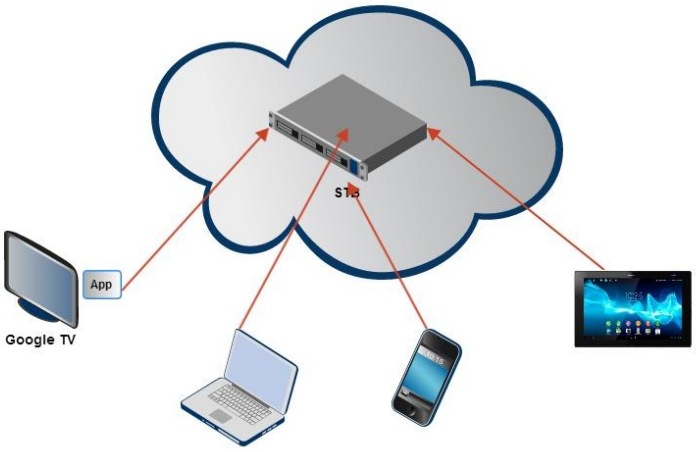

Figura 9: Acceso multidispositivo

Por consiguiente debemos programar nuestra aplicación en un lenguaje que sea compatible con la mayoría de los SO y browser, para así tener que hacer las mínimas modificaciones para adaptarlo a los diferentes entornos.

El funcionamiento del escenario básico consiste en un receptor de contenido (Smart TV) el STB virtual y un solo proveedor de IPTV. En la figura siguiente se puede ver gráficamente como funciona este sistema.

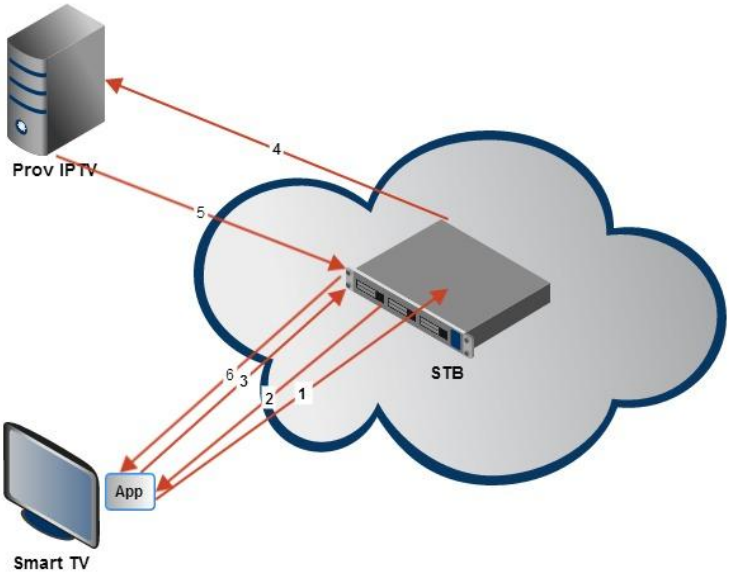

Figura 10: Escenario Básico

A continuación se detallan cada una de las flechas de la figura anterior:

1. El usuario por medio de una aplicación en su Smart TV (o televisión convencional a la que se pueda conectar un STB con SO Android) accederá a su aplicación.
2. El STB virtual le muestra la información almacenada en el servidor del proveedor de IPTV.
3. Discernir qué contenido desea visualizar, cuando el usuario decide, pulsa la opción de comprar.
4. EL STB virtual contactará con el proveedor de IPTV informándole que un usuario determinado quiere acceder a un contenido.
5. El proveedor comprobará si el usuario dispone de subscripción para comprarlo, es un sistema (Pay-as-you-go) para comprarlo, y en caso afirmativo el servidor le mandará una trama MPEG-2.
6. El STB virtual deberá ser capaz de decodificarlo antes de ser enviado al televisor para mostrarlo en la pantalla.

Este escenario tiene una serie de ventajas e inconvenientes:

- **Ventajas:**
  - No es necesario ningún hardware adicional para poder usar el servicio.
  - Usuario puede visualizar contenidos sin necesidad de haber previamente comprado el producto.
  - Multidispositivo.
- **Inconvenientes:**
  - Se necesita una Smart TV o STB con SO integrado.

3.2.2 Aprovechamiento

Las aplicaciones OTT no garantizan la entrega del contenido, ya que el estado de la red y la reserva de recursos son transparentes a este tipo de aplicaciones. Además son muy ineficientes debido a que realizan multitud de solicitudes HTTP para poder acceder a un mismo contenido.

En cambio con el STB virtual, si elegimos un contenido del que es propietario nuestro operador, se puede emitir en multicast ahorrando recursos de la red, frente a las aplicaciones OTT que emiten broadcast.

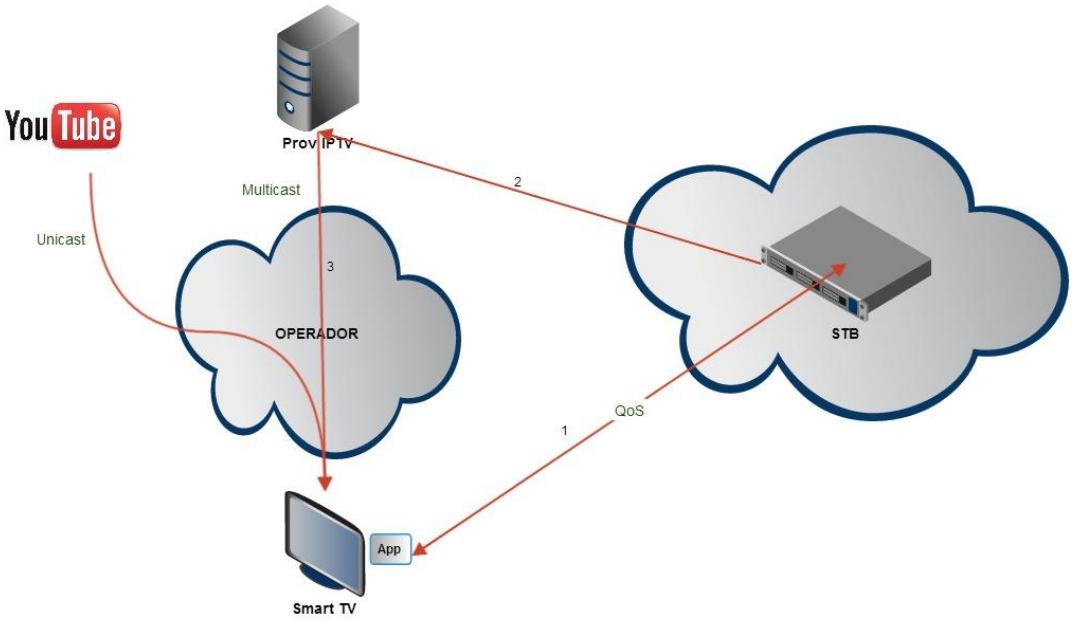

Figura 11: Aprovechamiento de la red

En la figura anterior se puede ver claramente el aprovechamiento mencionado. Youtube representa a las aplicaciones OTT. Los pasos que sigue el diagrama son los siguientes:

1 El Smart TV negocia la QoS con el STB virtual.

2 El STB virtual envía al Proveedor IPTV la información del subscriptor y del contenido al que desea acceder.

3 El Proveedor reserva recursos de su red en función del QoS negociado y emite en multicast.

Este escenario tiene ventajas significativas, pero también algún inconveniente.

- **Ventajas:**
  - Emisión multicast.
  - Posibilidad de negociar un QoS, a través de la aplicación, que permite evaluar retardos, realizar el control de la admisión de flujos y otras operaciones para calibrar la red.
- **Inconvenientes:**
  - Este escenario sólo es posible si el operador del usuario es el mismo que el proveedor del contenido que quiere visualizar.

3.2.3 Varios Proveedores

Una de las novedades de STB virtual es que se puede acceder al contenido de varios proveedores sin necesidad de tener una subscripción con cada uno por separado. Estos proveedores facilitarán su contenido al STB virtual, que será en el encargado de presentar el contenido de una forma ordenada y atractiva al usuario final. Esto es muy beneficioso para el usuario, ya que podrá tener una visión global de todos los contenidos, sin necesidad de buscar en cada proveedor IPTV por separado.

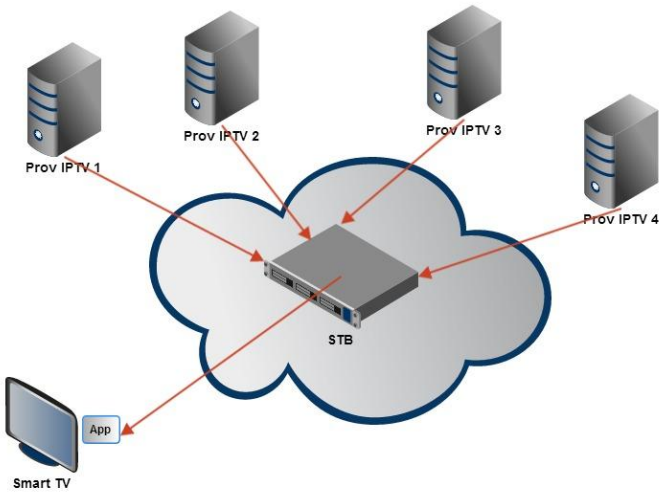

Figura 12: Escenario varios proveedores

En la figura anterior podemos ver gráficamente la descripción de este escenario. Cabe destacar que el proveedor IP podrá ser de cualquier país.

- **Ventajas:**
  - Accesibilidad a cualquier proveedor IPTV.
  - Visión global del contenido.
- **Inconvenientes:**
  - Los proveedores IPTV normalmente tienen muchos canales en común, este escenario será perjudicial para ellos.

3.2.4 Libertad de elección Video-On-Demand

Actualmente el usuario se subscribe a un proveedor IPTV y este es el que le proporciona el contenido al coste que desea. Este funcionamiento hace que el usuario no pueda comparar precios y por lo tanto no haya competencia a la hora de fijar el precio.

En este escenario, el cliente puede comparar el precio del mismo video en diferentes proveedores IPTV. Esto puede repercutir en una rebaja de los precios del contenido.

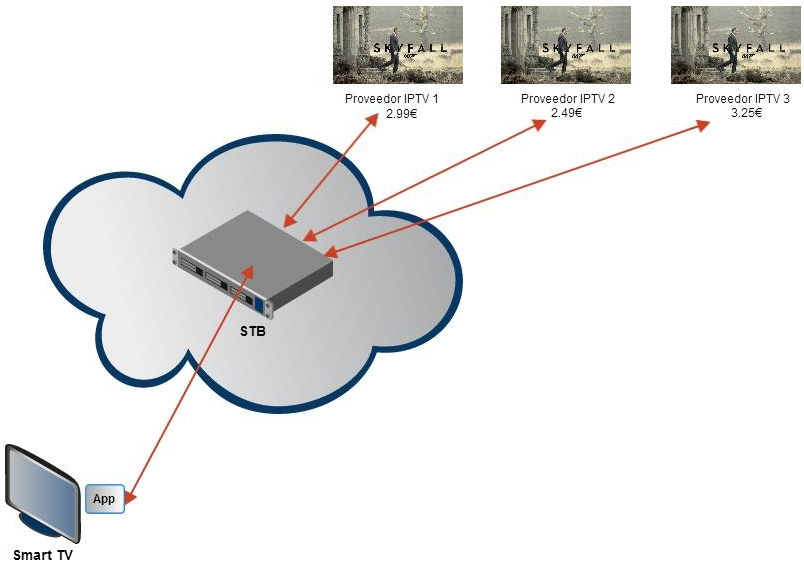

Figura 13: Escenario de elección VoD

En la figura anterior se muestra cómo al STB virtual le llegan el mismo video con 3 precios distintos provenientes de 3 proveedores.

- **Ventajas:**
  - El usuario puede comparar y elegir el precio más bajo.
- **Inconvenientes:**
  - Los proveedores de IPTV saldrían perdiendo puesto que deberían bajar los precios.

3.2.5 CAM Programable

Otro posible escenario es aprovechar un CAM programable, el cuál iría conectado a la Smart TV. Este CAM se comunica con el CI cuya información viajará cifrada por el algoritmo CSA, el cual tiene una clave que cambia en función del proveedor que le proporcione el contenido. Esta CAM programable deberá ser reprogramada por algún plug-in que contenga el Smart TV para que sea capaz de proporcionar el acceso condicional en función de donde provenga el contenido.

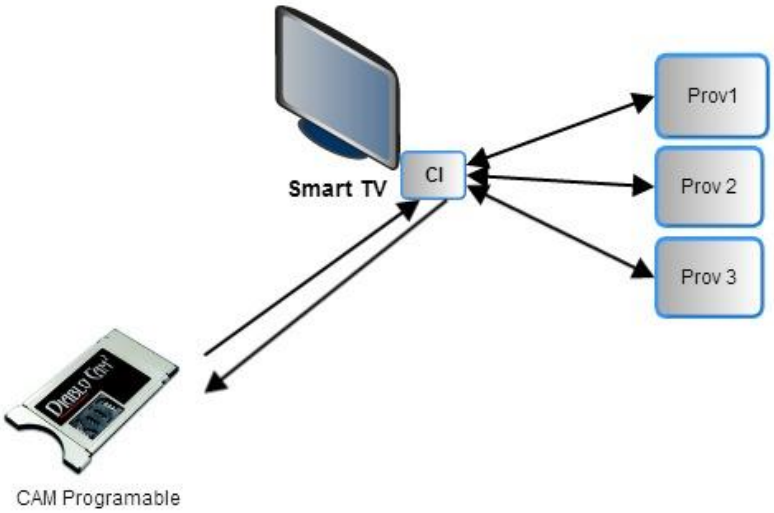

Figura 14: Escenario CAM programmable

En la figura anterior se puede ver el funcionamiento del escenario detallado anteriormente.

- **Ventajas:**
  - Misma CAM para distintos proveedores.
  - Mejora de seguridad a la hora del acceso a múltiples proveedores, vía hardware.
- **Inconvenientes:**
  - La aplicación debe ser capaz de programar CAMs de varios fabricantes.

La mayoría de los escenarios propuestos, no son independientes, sino que se pueden combinar entre ellos, complementando el escenario básico. Tras este análisis, en el siguiente apartado resumiremos las ventajas e inconvenientes del sistema en general, una vez que el lector tiene conocimiento de las ventajas de cada uno de los escenarios.

3.3 Resumen de ventajas e inconvenientes

- **Ventajas:**
  - Bajo coste y alta actualización.
  - Accesibilidad a todo el contenido sin necesidad de subscripciones.
  - Uso de los recursos de la red más eficiente.
  - CAM programable mejora la seguridad en el acceso a múltiples proveedores.
  - Mantiene la personalización entre proveedores.

- **Inconvenientes:**
    - Demasiado contenido puede abrumar al cliente si no se ordena y gestiona de una forma adecuada.
    - Los proveedores IPTV perderían su posición en el mercado nacional, al entrar en el libre mercado, compitiendo con otros proveedores IPTV de todo el mundo, pero podrían también usuarios extranjeros acceder a él.
    - Se difumina la geolocalización.

3.4 Retos

En este apartado vamos a enumerar y a realizar un breve análisis de los principales retos que el proyecto se va a encontrar teniendo en cuenta el estado del arte actual.

- Conseguir que una aplicación de un Smart TV reproduzca videos usando una tecnología común a la mayoría de los dispositivos existentes en el mercado.
- Implementación del STB virtual con una tecnología común a la mayoría de los dispositivos.
- Streaming Live, uno de los objetivos de nuestro STB virtual es que el usuario pueda acceder a canales en directo, para ello se estudiará qué protocolo de los existentes es mejor en nuestro proyecto.
- Compatibilidad con los dispositivos, habrá que intentar que el STB virtual sea compatible con la mayoría de dispositivos. Si existiera alguna incompatibilidad, se propondrá una solución para que la aplicación sea válida en la mayoría de ellos.

En resumen, el reto principal de nuestro proyecto es conseguir que sea accesible desde la mayor parte de los dispositivos. Para ello se estudiará que tecnologías son las más idóneas para alcanzar este objetivo.

Capítulo 4

Diseño

4.1 Introducción

En el presente capítulo se pretende proporcionar al lector una visión de cómo se va a realizar el concepto descrito en el capitulo anterior, además se va a especificar partes del diseño en las que se ha realizado un análisis detallado.

Al finalizar el capitulo actual el lector tendrá una visión clara tanto de qué es el STB virtual y de los posible escenario que se pueden implementar, los cuales se describieron en el capitulo anterior, como de la arquitectura y desarrollo de los diferentes elementos que la forman.

4.2 Arquitectura modular

En esta sección vamos a describir la arquitectura modular del sistema, esta se puede visualizar en la siguiente figura la cual se mostró en el capítulo anterior. En ella se puede distinguir varios elementos como son el proveedor de televisión IP, el STB virtual y el Smart TV.

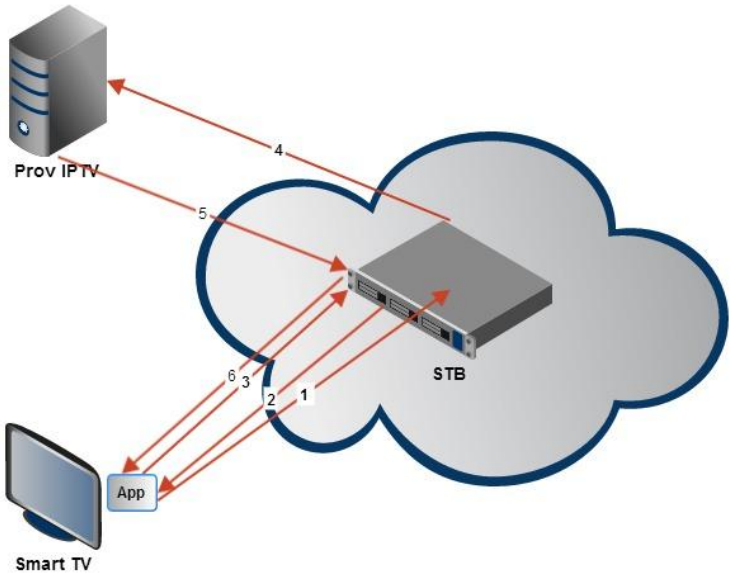

Figura 15: Arquitectura modular básica

- **Proveedor IPTV**

El proveedor IPTV principalmente es el encargado de proporcionar el contenido multimedia que se reproducirá en la aplicación. Pero también se encarga de otras funciones como:

  - Adaptar el contenido multimedia al dispositivo seleccionado. Esta adaptación como está descrito en el estado del arte puede ser mediante una codificación escalable (H.264/SVC) en la cual se establece un nivel de calidad básico y si según se añadan niveles irá mejorando la calidad del video. La otra opción es realizar una transcodificación del contenido, por el cual se logra adaptar el contenido a los formatos y resoluciones de los dispositivos finales.
  - Controlar datos de las sesiones de los subscriptores, tales como el crédito o el contenido al cual puede acceder.

El proveedor de IPTV además de tener estás funciones necesita ofrecer un streaming de alta velocidad para que el usuario pueda disfrutar del contenido sin cortes.

- **STB virtual**

Es el eje del proyecto, el cual se describió detalladamente en el capítulo 3, será el encargado de hacer de intermediario entre el Proveedor IPTV y la aplicación de la Smart TV.

- **Smart TV**

En las Smart TV estará instalada la aplicación, la cual dará el soporte necesario para garantizar la seguridad del contenido y negociará una QoS con el STB virtual.

Además deberá presentar el interfaz del contenido, debe ser agradable a la visión del usuario y optimizado para este entorno.

4.3 Arquitectura funcional

El STB virtual va a tener la siguiente arquitectura funcional, los diferentes módulos que la componen se detallarán posteriormente:

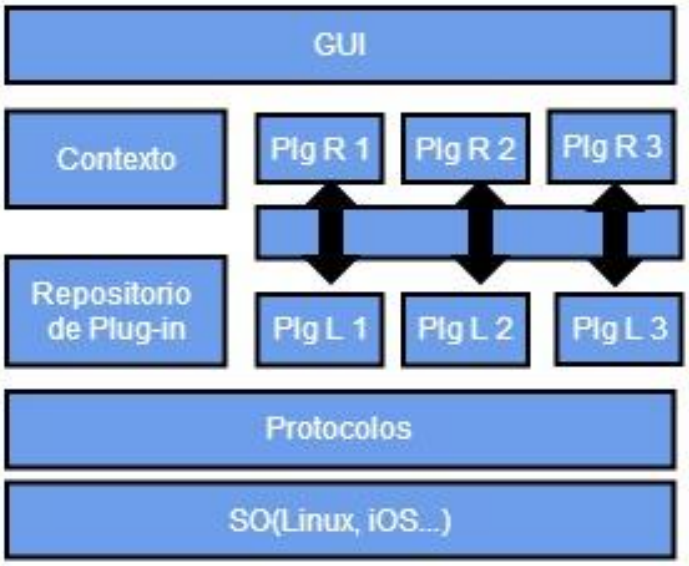

Figura 16: Arquitectura funcional del STB virtual

4.3.1 Descripción de bloques

Para describir los bloques de la arquitectura funcional vamos a empezar por el bloque superior para luego ir descendiendo hasta el último.

En lo alto de la torre tenemos un nivel encargado del interfaz gráfico del usuario (GUI). El principal uso de este nivel es proporcionar un entorno visual que permita con el SO de la Smart TV. Este nivel es el que permite instalar software y aplicaciones en el STB virtual.

En el nivel intermedio podemos distinguir entre los plug-in y el contexto.

- **Plug-in**

  En la arquitectura funcional diseñada hemos incluido varios plug-ins de dos tipos, locales y remotos. Esta división de plug-ins en remoto y local permite poder separar físicamente el nivel de interfaz de usuario del resto, pero manteniendo la conectividad entre ellos. Los plug-ins se forman por pares, y cada plug-in local tiene un remoto asociado. El plug-in remoto se puede comunicar directamente con el nivel superior, el GUI, en cambio el plug-in local se puede comunicar con los niveles inferiores, la pila de protocolos y el SO. Adicionalmente los plug-in remotos y locales se comunicarán entre ellos para poder intercambiar la información obtenida o hacer una petición para obtener alguna información.

En el siguiente apartado describiremos más detalladamente la distribución de los plug-in y su ubicación dentro de la arquitectura modular del STB virtual.

Para terminar con este apartado el bloque de Repositorio de plug-ins que es donde se encuentran realmente los plug-ins, desde este bloque es donde se accede a los plug-in, lo cuales se pueden instalar, desinstalar o actualizar.

- **Contexto**

  Este bloque almacena la información sobre configuración y personalización del usuario. Esta información abarca desde el estilo de la fuente, hasta la gama de colores utilizados. En la actualidad, cada proveedor IPTV tiene su propio contexto, pero carecería de sentido aplicar un estilo diferente para mostrar el contenido de cada proveedor. Por estos motivos, se podrá utilizar cualquiera de los contextos utilizados en cualquiera de los proveedores IPTV a los que se tenga acceso.

Inmediatamente debajo del contexto y los plug-in encontramos una pila de protocolos, esta pila de protocolos está proporcionada por el Smart TV y puede ser usada por los niveles superiores.

Finalmente, en el nivel más bajo encontramos el SO, el cual podrá ser Android, iOS o cualquier otro, el cual forma parte del Smart TV.

4.3.2 Distribución de los bloques

Antes de estudiar la distribución de los bloques de la arquitectura funcional, conviene aclarar al lector, que la Smart TV proporciona los niveles inferiores (SO y pila de protocolos) de la arquitectura funcional de STB descrito en el apartado anterior. Adicionalmente, debido a las características de la nube se pueden implementar los demás niveles de esta arquitectura, lo que permite emular el funcionamiento de un STB tradicional con la ayuda de la Smart TV y la nube.

Una vez se tienen estos conceptos claros, se va a mostrar en la siguiente figura la solución adoptada. Posteriormente se razonará todas las decisiones de diseño de los mismos, así como algunos aspectos de su funcionalidad.

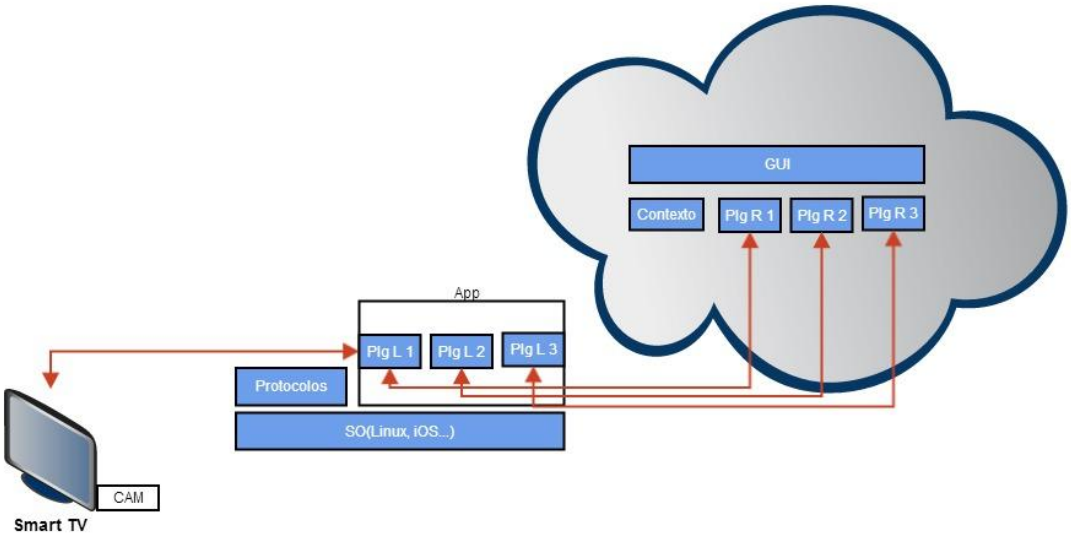

Figura 17: Arquitectura mixta Funcional vs Modular

La figura anterior, es un híbrido de las arquitecturas funcional y modular. En ella como se adelantó anteriormente, se ha dividido en dos partes la arquitectura funcional, dejando la comunicación entre plug-ins locales y remotos como nexo entre ambas.

Se va a analizar por separado el módulo que se ha quedado en la Smart TV, el alojado en la nube y su interacción.

4.3.2.1 Módulos en la Smart TV

Los niveles correspondientes a la pila de protocolo y el SO vienen incorporado en las Smart TV, como se explicó en el estado del arte. Adicionalmente se le ha incorporado los plug-ins locales. Estos plug-in locales serán informados de cualquier acción que realice el usuario, y decidirán si intervienen o no, dependiendo de su función.

Algunos de los plug-ins más comunes en este tipo de dispositivo son: EPG, protección de contenido, seguridad, mantenimiento de la sesión, entrega del contenido, entre otros. Para que el lector no tenga una visión tan abstracta de este sistema de plug-in vamos a comentar algunos ejemplos que se podrían llevar a cabo gracias a este sistema y sus ventajas. Para ello nos centramos en los plug-in de EPG, protección de contenido y seguridad:

- EPG: es la guía electrónica de programas de la televisión, en la que se ordena de forma sencilla los canales y además se introduce una información adicional. La EPG es proporcionada por cada proveedor IPTV, por lo que si se opta por una implementación

con múltiples proveedores, figura 12, habrá múltiples EPGs. En este escenario será la aplicación del Smart TV la encargada de juntar cada una de las EPGs.

- Protección de contenido: se puede dotar al plug-in local de una serie de reglas de uso. Esto permitiría poder distribuir el video de forma adecuada al DRM sin necesidad de consultar al proveedor IPTV. Sólo consultaría al plug-in remoto para saber datos de los usuarios como el estado de la subscripción.
- Seguridad: En este bloque se proporcionará funcionalidades necesarias para aumentar la seguridad del contenido. Estas funcionalidades pueden ir desde una mejora software hasta una comunicación con el CAM para poder usar el estándar DBV-CA y poder mejorar la seguridad del contenido. Concretamente en el escenario de una CAM programable, figura 14, el plug-in local podría tener el algoritmo para reprogramar la CAM, y el plug-in remoto le pasaría la información del proveedor IPTV para poder programarla.

4.3.2.2 Módulos en la nube

Los plug-in remotos se han explicado en el apartado anterior junto a los locales. En este apartado, vamos a razonar porque se ha escogido la nube, que necesitamos que aporte para que pueda funcionar el STB virtual. Para empezar, vamos a describir brevemente las diferentes soluciones Cloud Computing. Estas soluciones se pueden dividir en tres tipos:

- **Infraestructure as a Service (IaaS)**
  Este modelo contratamos **capacidad de proceso (CPU) y almacenamiento**.
  Normalmente se accede mediante una maquina virtual. Algunos de los más conocidos son EC2 de Amazon y GoGrid.
- **Platform as a Service (PaaS)**
  Este modelo ofrece todo lo necesario para soportar el ciclo de vida completo de construcción y puesta en marcha de aplicaciones y servicios web completamente disponibles en la Internet. Por ejemplo, Google App Engine y Azure de Microsoft.
- **Software as a Service (SaaS)**
  Este modelo es el que normalmente identifica con “cloud”. Es una aplicación para el usuario final donde paga un alquiler por el uso de software. Por ejemplo: GoogleDocs, Zoho o Office365.

Debido a que el STB virtual requiere procesamiento tanto para lo plug-in remotos como para el GUI, PaaS o IaaS podrían servir a nuestros propósitos. Cualquiera de los dos sería válido para este modelo.

4.3.2.3 Interacción entre ambos

Como se describió anteriormente los pares de plug-in deberán poder comunicarse entre sí, para ello pueden usar cualquier protocolo que lo permita. Dos posibles implementaciones son REST y SOAP.

Figura 18: Intercambio mensajes

Para nuestro STB virtual implementaremos REST debido a que se puede usar JSON, y este lenguaje es idóneo para dispositivos con pocos recursos, como es la Smart TV.

Las ventajas que nos ofrece JSON frente a XML:

- JSON es la sintaxis para almacenar e intercambiar información de texto. Al igual que XML.
- JSON es más ligero que XML, y, por tanto, más rápido de transmitir y más fácil de analizar.

Capítulo 5

Estudio de compatibilidades y prueba de concepto

5.1 Introducción

En este capítulo vamos a analizar detenidamente si se pueden implementar todas las funcionalidades del proyecto o si hay que reducir las expectativas debido al desarrollo de las mismas. Además se implementará una prueba de concepto del mismo.

Para empezar, se ha decidido implementar con el lenguaje HTML5 debido a que es un lenguaje de prestaciones adecuadas para nuestros propósitos, ya que es compatible con multitud de plataformas. Algunas de sus ventajas son:

- **Mejor visibilidad en contenidos móviles**.
- **Estructura del cuerpo**: La mayoría de las webs tienen un formato común, formado por elementos como cabecera, pie, navegadores, etc. HTML 5 permite agrupar todas estas partes de una web en nuevas etiquetas que representarán cada uno de las partes típicas de una página.
- **Bases de datos locales**: el navegador permitirá el uso de una base de datos local, con la que se podrá trabajar en una página web por medio del cliente y a través de un API. Es algo así como las Cookies, pero pensadas para almacenar grandes cantidades de información, lo que permitirá la creación de aplicaciones web que funcionen sin necesidad de estar conectados a Internet.
- **Web Sockets**: Proporciona una forma mucho más rápida de transferencia de datos en línea y comunicación, pudiendo recibir actualizaciones en tiempo real desde servidores.
- **Mejor acceso sin conexión**: con aplicaciones ricas de Internet que almacenarán más información, como mensajes de correo electrónico para ver incluso sin conexión.
- **Ubicación geográfica**: posibilidad de ubicar al usuario que navega el sitio.
- **Sin plug-ins**: Con HTML5 además es posible reproducir de forma nativa en el navegador audio y video.
- **Fin de las etiquetas de presentación**: todas las etiquetas que tienen que ver con la presentación del documento, es decir, que modifican estilos de la página, serán

eliminadas. La responsabilidad de definir el aspecto de una web correrá a cargo únicamente de CSS.

Por otra parte, para el análisis de compatibilidad vamos a dividirlo en 2 bloques, por un lado compatibilidad con HTML5 y sus videos, por el otro la compatibilidad con la visualización de canales en directo y las compatibilidades con los navegadores actuales.

Algunos de los protocolos actuales para distribuir el contenido multimedia por internet son:

- HLS de Apple.
- Smooth Streaming de Microsoft.
- HDS de Adobe.

Para nuestro diseño hemos utilizado HLS, ya que ofrece una mayor compatibilidad con HTML5 y se adapta mejor a los requisitos del STB virtual y el Smart TV.

Por último en este capítulo se explicará la prueba de concepto que se ha implementado así como la justificación de la tecnología utilizada y el interfaz resultado.

Por otra parte como SO se ha elegido Android debido la gran popularidad actual y a que proporciona un SDK abierto. Android posee un Add-on o plug-in para la versión 3.1 y 3.2 que proporciona unas funcionalidades añadidas para mejorar su integración en televisiones, este plug-in es el llamado Google TV.

Debido a los plug-in usados en el STB virtual, se podría personaliza el plug-in para que fuera compatible con otras versiones de Android. Esto sería de gran utilidad debido a la creciente distribución de USB con Android incorporado y Smart TV con versiones de Android distintas a la 3.1 y 3.2.

5.2 Compatibilidad con videos en HTML5

Google TV usa actualmente Android 3.2, está versión es compatible con HTML5 en la mayoría de sus especificaciones, concretamente para el objeto de este proyecto nos interesa que tenga compatibilidad con el tag <video> y con algún formato de video.

Se va a recoger en la siguiente gráfica las compatibilidades de video que ofrecen las últimas versiones de los navegadores más comunes con los diferentes formatos de video aceptados en HTML5.

Browser / Video	Chrome 29	Firefox 24	Google TV	Internet Explorer 11
video element	Yes ✔	Yes ✔	Yes ✔	Yes ✔
Subtitle support	Yes ✔	No ✘	No ✘	Yes ✔
Poster image support	Yes ✔	Yes ✔	Yes ✔	Yes ✔
MPEG-4 support	No ✘	No ✘	Yes ✔	No ✘
H.264 support	Yes ✔	No ✘	Yes ✔	Yes ✔
Ogg Theora support	Yes ✔	Yes ✔	No ✘	No ✘
WebM support	Yes ✔	Yes ✔	No ✘	No ✘

Tabla 1: Compatibilidades de los browser con HTML5

Como podemos apreciar, actualmente no hay un formato aceptado por todos los navegadores más comunes, lo cual no es un problema crítico en HTML5 ya que el tag <video> permite hacer referencia a diferentes fuentes en diferentes formatos, y cada navegador cargará el que sea compatible. Para implementar un video en HTML5 se deberá introducir el siguiente código:

```
<video width="320" height="240" controls>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
Your browser does not support the video tag.
</video>
```

Con este nuevo Tag de HTML5 se podría reproducir el video en todos los navegadores analizados, y si se reprodujera en un navegador que no es compatible que ningún formato, se vería en el display: “Your browser does not support the video tag”.

No obstante, el video tiene que ser previamente transcodificado, para adaptarlo a los formatos permitidos. Concretamente en este proyecto se han convertido los videos a todos los formatos soportados por HTML5 con el fin de proporcionar la mayor compatibilidad posible para cualquier browser que intente acceder al contenido.

En cuanto a la compatibilidad de cualquier browser con HTML5 tenemos que concluir que será completa, y que podremos reproducir el contenido en cualquier browser, por otro lado la compatibilidad con Google TV, teóricamente también debería ser buena, no obstante en el siguiente capítulo se realizarán las pruebas oportunas para verificar su funcionamiento.

5.3 Compatibilidad con canales en directo

Como hemos comentado anteriormente se usará HLS, este sistema de Apple ofrece una compatibilidad completa con HTML5 y tanto su navegador Safari como su SO iOS son compatibles completamente con él. HLS goza de una gran popularidad en la actualidad, por esta razón Android ha hecho grandes esfuerzos para integrarlo.

A continuación ofrecemos una tabla donde se puede apreciar que prácticamente ningún buscador es compatible con esta solución.

Browser/Device	HTTP Live Streaming	Media Source API
Chrome	No	Yes
Firefox	No	No
Internet Explorer	No	No
Safari	Yes	No
iOS	Yes	No
Android	No (Yes 4.2+)	No
Opera	No	No

Tabla 2: Sistemas compatibles con HLS

Android afirma ser compatible con HLS concretamente las versiones Android 3.x con la versión 2 y Android 4.0 y superior con la versión 3. Pese a estas afirmaciones, lo cierto es que HLS no funciona bien en Android. En la mayoría de las versiones tiene problemas como que el reproductor se congela al intentar acceder a pantalla completa, no reproduce hasta que no se pulsa el botón de play 3 veces o al reproducirlo por primera vez el video solo suena el audio. Estos son entre otras, algunos de los problemas más usuales que han experimentado los usuarios con las versiones actuales de Android.

En el capítulo 6 se realizarán diversas pruebas sobre diversas sobre la compatibilidad de las diferentes versiones de Android y HLS.

5.4 Prueba de concepto

En cuando a la prueba de concepto la hemos dividido su implementación en dos partes:

- Desarrollo del servidor
- Desarrollo de la aplicación

A continuación se detallarán cada una de ellas, así como los resultados obtenidos, no obstante las conclusiones de esos resultados de dejan para el capítulo de pruebas.

5.4.1 Desarrollo del servidor

Para realizar las pruebas del STB virtual se ha desarrollado con JSF, utilizando un servidor Glassfish, se hace esta elección puesto que Glassfish es un servidor de aplicaciones multiplataforma de software libre. En el caso del proveedor de servicios podría usar la versión para empresas GlassFish Enterprise Server.

Por otra parte, el objetivo de la tecnología JavaServer Faces es desarrollar aplicaciones web de forma similar a como se construyen aplicaciones locales con Java Swing, AWT (Abstract windowToolkit), SWT (Standard Widget Toolkit) o cualquier otra API similar.

La tecnología JavaServer Faces constituye un marco de trabajo (framework) de interfaces de usuario del lado de servidor para aplicaciones web basadas en tecnología Java y en el patrón MVC (Modelo Vista Controlador).

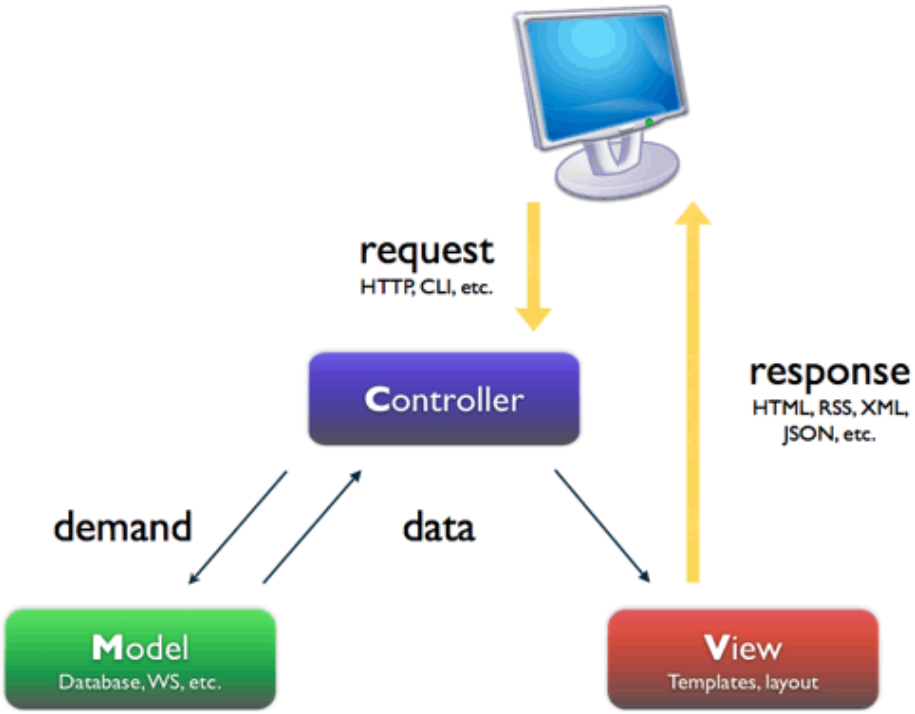

Figura 19: MVC [Tumaes]

Particularizando para nuestra aplicación, vamos a hacer un breve análisis de lo incluido en cada uno de los bloques:

- **Modelo**: En este apartado se ha implementado un conjunto de clases en java que conociendo en que carpetas se encuentra el contenido (los diferentes formatos de

videos, el poster, etc) es capaz de presentar el contenido en la página web independientemente de que el proveedor vaya añadiendo o quitando elementos.

- **Vista**: Para la vista se han implementado unas plantillas que posteriormente se han usado para presentar el contenido. Se ha utilizado CSS3, que ofrece una gran variedad de opciones muy importantes para hacer atractivo el contenido al usuario.
- **Controlador**: El controlador hace de intermediario entre la vista y el modelo de tal forma que adapta la vista si el usuario hiciera una operación que lo requiriese.

Los principales componentes de la tecnología JavaServer Faces son:

- **Similitud con HTML**: El código JSF con el que creamos las vistas (las etiquetas jsp) es muy similar al HTML. Lo pueden utilizar fácilmente desarrolladores y diseñadores web.
- **Compatibilidad con JSP**: JSF se integra dentro de la página JSP y se encarga de la recogida y generación de los valores de los elementos de la página.
- **Mejora las operaciones internas**: JSF resuelve validaciones, conversiones, mensajes de error e internacionalización.
- **Integración de javascript**: JSF permite introducir javascript en la página, para acelerar la respuesta de la interfaz en el cliente (navegador del usuario).
- **Tecnología extensible**: por lo que se pueden desarrollar nuevos componentes a medida, También se puede modificar el comportamiento del framework mediante APIs que controlan su funcionamiento.

Los contenidos se han alojado en un servidor Apache Tomcat, podrían haberse almacenado también en el mismo servidor Glassfish, pero se separan para simular la separación entre el STB virtual y el servidor del proveedor de contenidos.

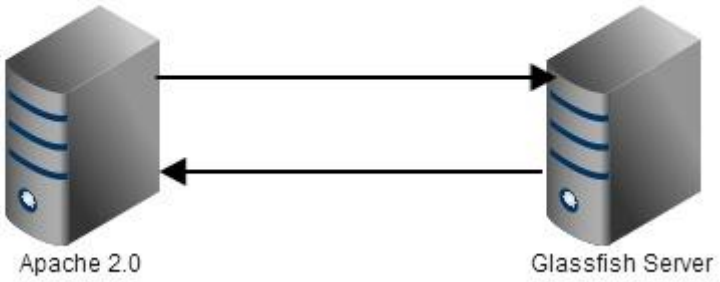

Figura 20: Comunicación del contenido

En cuanto al interfaz de usuario de STB virtual, se ha optado por un fondo oscuro, con el fin de resaltar las imágenes sin que la pantalla tenga una excesiva iluminación que podría ser desagradable a la vista.

Para dividir el contenido se ha utilizado una plantilla de división en tabulaciones, proporcionada JQuery UI. Se elije JQuery UI porque es una biblioteca Javascript que simplifica la interacción con HTML y además es compatible con Android.

En nuestra aplicación en concreto disponemos de una serie de contenidos que dividiremos según su tipo en películas, series y documentales.

El interfaz principal tiene este aspecto:

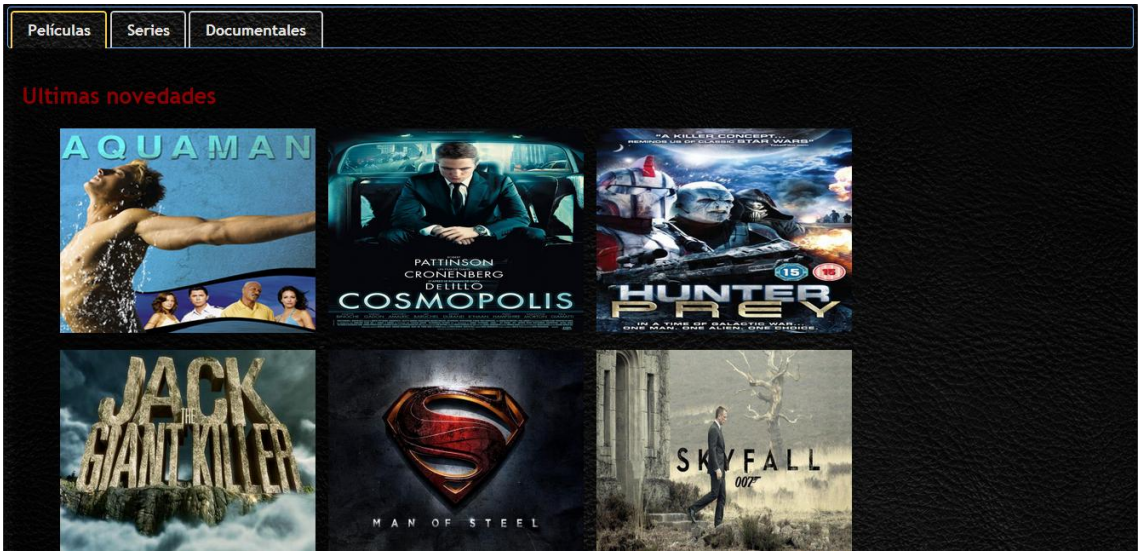

Figura 21: Interfaz principal

Cuando pulsamos sobre una de las imágenes, nos lleva directamente al contenido, para mejorar el interfaz del video se ha introducido un reproductor. El resultado es el siguiente:

Figura 22: Interfaz del reproductor de video

Como se puede apreciar el fondo no es oscuro para proporcionar poca luz y que el contenido pueda ser visualizado correctamente, no obstante el usuario podrá reproducirlo en pantalla completa.

5.4.2 Desarrollo de la aplicación

Una vez realizada completamente el interfaz de usuario que presta el STB, queremos crear una aplicación que muestre el contenido de una forma adaptada al dispositivo en el que se está realizando.

Las Smart TV poseen SO que permite a los usuarios descargar aplicaciones e instalarlas en sus dispositivos. En nuestro caso Android tiene el Google Play, desde cual se puede acceder a todo tipo de aplicaciones, entre las cuales podría estar la que se comunica con el STB virtual.

Aprovechando que se ha implementado el interfaz del STB, la aplicación deberá ser capaz de adaptar el interfaz del STB a las dimensiones y características del dispositivo.

Como razonamos al principio del capítulo, se ha elegido el SO Android. Android dispone de un método para realizar esta adaptación llamado WebView. Para programar nuestra aplicación nos hemos basado en un ejemplo ubicado en Google Code, el cual es un entorno gratuito de software libre. Este ejemplo propone la utilización de un layout como el que se muestra en la siguiente figura.

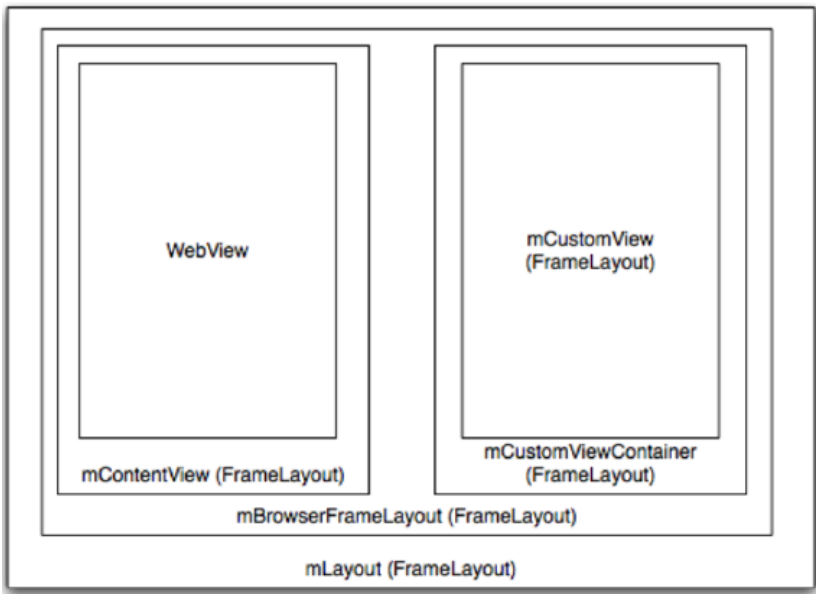

Figura 23: HTML5 webView Layout [tandroidHTML5]

La utilización de este layout nos va a ofrecer la posibilidad de poder visualizar la pagina en HTML5 y sus videos. En cuanto a la funcionalidad hemos usado dos subclases que nos

proporciona WebView y que son de gran utilidad para poder realizar las funciones del STB virtual.

- **WebChromeClient**: esta clase es llamada cuando ocurre algo que podría impactar al interfaz del navegador, por ejemplo, las alertas de JavaScript son enviadas aquí.
- **WebViewClient**: esta clase será llamada cuando ocurren acciones que impactan la representación del contenido, p.ej. errores. También se puede interceptar URL cuando estas están cargando (con el método: shouldOverrideUrlLoading()). Esto será de gran utilidad para la funcionalidad de los plug-ins que detallaremos posteriormente.

Con ayuda de estos dos métodos se va a implementar la clase que ayudará a la correcta visualización de los videos.

En la figura siguiente se puede ver como se reproduce correctamente un video en la aplicación:

Figura 24: Captura de pantalla de la aplicación en un móvil Android 4.1

A parte del interfaz del usuario y de poder reproducir videos y navegar por ellos, se necesita poder realizar otras operaciones que dependerán de cómo va navegando por la aplicación nuestro usuario. Para implementar esta funcionalidad haremos uso de plug-ins, lo cual se detalló en el capítulo 4.

Para dar una idea al lector del cómo se comunican los plug-ins, se ha realizado una imagen que se describirá posteriormente:

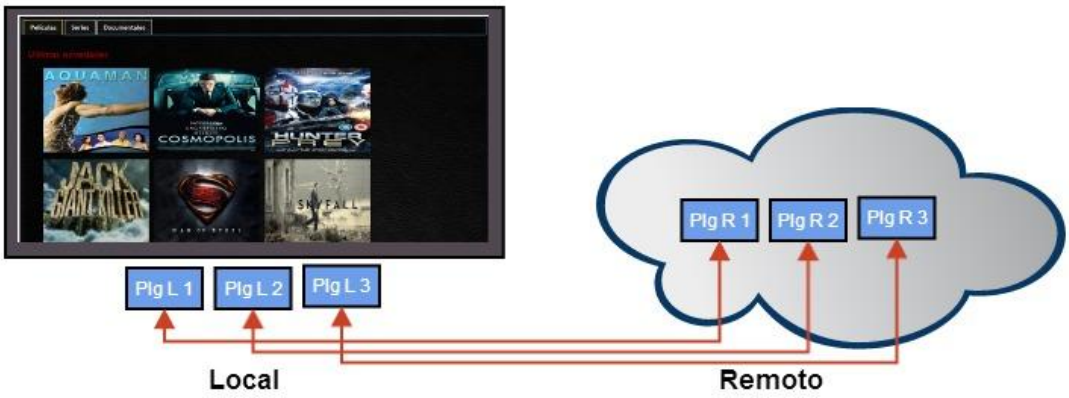

Figura 25: Uso de Plug-in

Aunque no aparece en la figura anterior hay un repositorio de plug-ins los cuales pueden ser instalados o desinstalados. Los plug-ins que se muestran en la figura 25 están instalados. Los plug-ins locales al detectar un cambio, pueden comunicarse directamente con el proveedor IPTV o con su correspondiente plug-in remoto, el cual está en la nube.

Para comprobar que el sistema de plug-ins funciona, hemos añadido la siguiente funcionalidad a nuestra aplicación. Se explicará con atención el flujo del programa, apoyado en el siguiente diagrama:

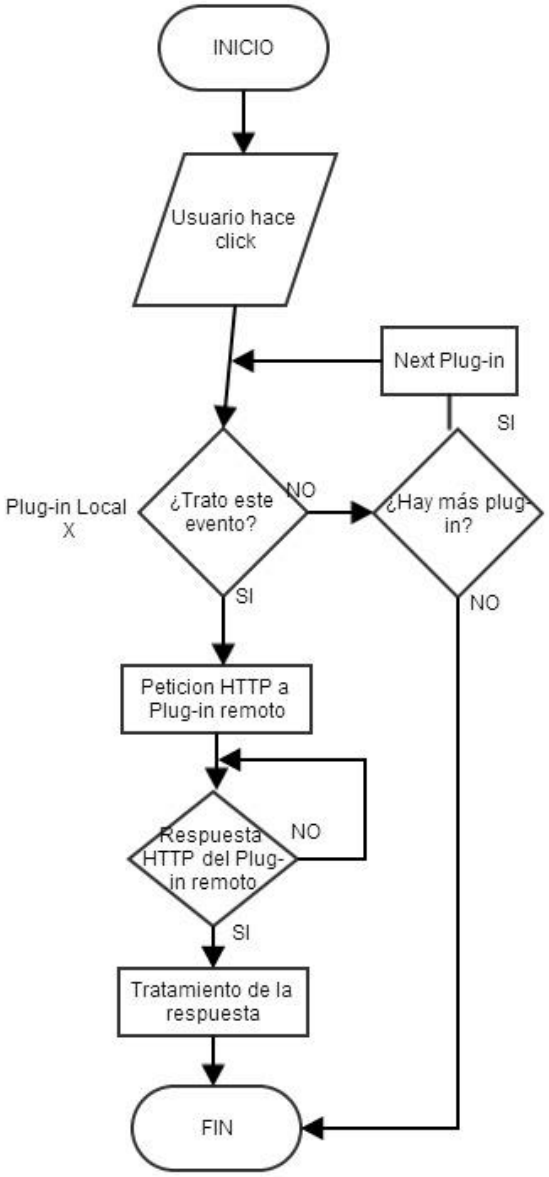

Figura 26: Diagrama de flujo de un evento

Cuando el usuario hace “click” sobre un video, se informará a los plug-ins locales del evento, si alguno le afecta el evento, este se activará. En concreto para nuestra funcionalidad el plug-in local mandará una petición HTTP a su plug-in remoto asociado. En esta petición HTTP el plug-in local preguntará al plug-in remoto de que proveedor IPTV es el video que quiere ver el usuario. Lo cual se traduce en una petición HTTP acerca del contexto, al cual puede acceder el plug-in remoto. Esta implementación es una de las muchas que se podrían llevar a cabo. Gracias a los plug-ins se puede tener control sobre todo lo que el usuario realiza, ayudando a mejorar la experiencia de usuario.

Capítulo 6

Pruebas

6.1 Introducción

En este capítulo se detallan las pruebas que se han ido realizando para demostrar la posibilidad de implementar el STB virtual, y los resultados que se han obtenido.

En primer lugar se realizó una prueba para verificar el correcto funcionamiento del emulador de Google TV, para ello se realizó una simple aplicación que reproducía un video a través de una página subida a un servidor apache en HTML5.

Posteriormente se creó el interfaz de usuario del STB virtual, cuyo funcionamiento se detalla en el capítulo anterior. A su vez se desarrollan diferentes pruebas sobre él y sobre su compatibilidad con Google TV.

Como prueba adicional también se realizó una prueba con HLS tanto de VoD como de televisión en directo.

Para presentar las pruebas y sus resultados se han realizado unas tablas compuestas con las siguientes columnas:

- Prueba realizada: Describe el dispositivo o versión de Android utilizado.
- Resultado: Puede ser OK, NO OK o ~OK. Este último significa que hubo algún problema, pero tuvo algunas funcionalidades.
- Observaciones: Comentarios acerca de las pruebas.

6.2 Prueba de reproducción de video sobre Google TV

Esta prueba consiste en desarrollar una sencilla web en HTML5, la cual contiene un video. A este video se accederá desde una aplicación Android. El objetivo es poder acceder desde el emulador de Google TV.

Adicionalmente se han realizado pruebas en otras versiones de Android y también con una en una Smart TV. En la siguiente tabla podemos observar las pruebas realizadas:

Prueba realizada	Resultado	Observaciones
Emulador Google TV 3.1 en Ubuntu 12.04	OK	Se pudo reproducir un video perfectamente y sin cortes
Emulador Google TV 3.2 en windows 7	NO OK	No se pudo reproducir ningun video, el emulador se queda colgado.
Emulador Android 4.3	~OK	El sonido se reproduce sin ningun corte, pero el video va a saltos. Sólo se pudo reproducir el video en pantalla completa.
LG Smart TV	OK	Se reproduce el audio y el video con buena calidad (usa el formato webm) y es capaz de reproducirlo normal y en pantalla completa.
Smartphone con Android 4.1	OK	El video HTML5 se pudo reproducir correctamente sin ninún corte tanto en pantalla completa como reducida.

Tabla 3: Prueba video HTML5

Para que funcionen los videos en todos los dispositivos es necesario convertirlos a los formatos soportados por HTML5, para que dependiendo del dispositivo reproduzca uno u otro.

6.3 Prueba del STB virtual

Una vez desarrollado el STB virtual, se han realizado una serie de pruebas para garantizar su funcionamiento. Se realizaron pruebas de acceso desde navegadores en distintos dispositivos. En la siguiente tabla se recogen los resultados.

Prueba realizada	Resultado	Observaciones
Acceso desde distintos navegadores: Firefox 24, Chrome 29, IE10	OK	En todos los navegadores se pudo mostrar adecuadamente el contenido.
Acceso desde una TV LG Smart TV	OK	El contenido se muestra perfectamente. Dificultad para moverse debido al mando debido a la falta de adaptación. Por el contrario el contenido se carga rapidamente.
Acceso desde dispositivo movil Android 2.3.6	NO OK	La pagina principal se muestra perfectamente, pero no es posible acceder a los videos, debido a que su explorador no es compatible con los formatos soportados por HTML5
Acceso desde elnavegador del emulador Google TV	NO OK	El emulador de Google TV a demás de su lentitud, tiene el mismo problema que la prueba anterior.
Acceso desde dispositivo movil Android 4.1.2	OK	Desde este dispositivo se pudieron reproducir los videos usando el Chrome.

Tabla 4: Prueba interfaz STB virtual

Como cabía esperar después de haber realizado el estudio de compatibilidades todos los navegadores soportan HTML5 y el tag <video>. En contra de lo que se pensaba el emulador de Google TV no fue capaz de reproducirlo, esto puede es debido a que el emulador tiene ciertas limitaciones, es probable que en un dispositivo físico Google TV se pudiera acceder al contenido. En la siguiente imagen podemos apreciar como el Smart TV de LG pudo acceder al interfaz de usuario del STB virtual.

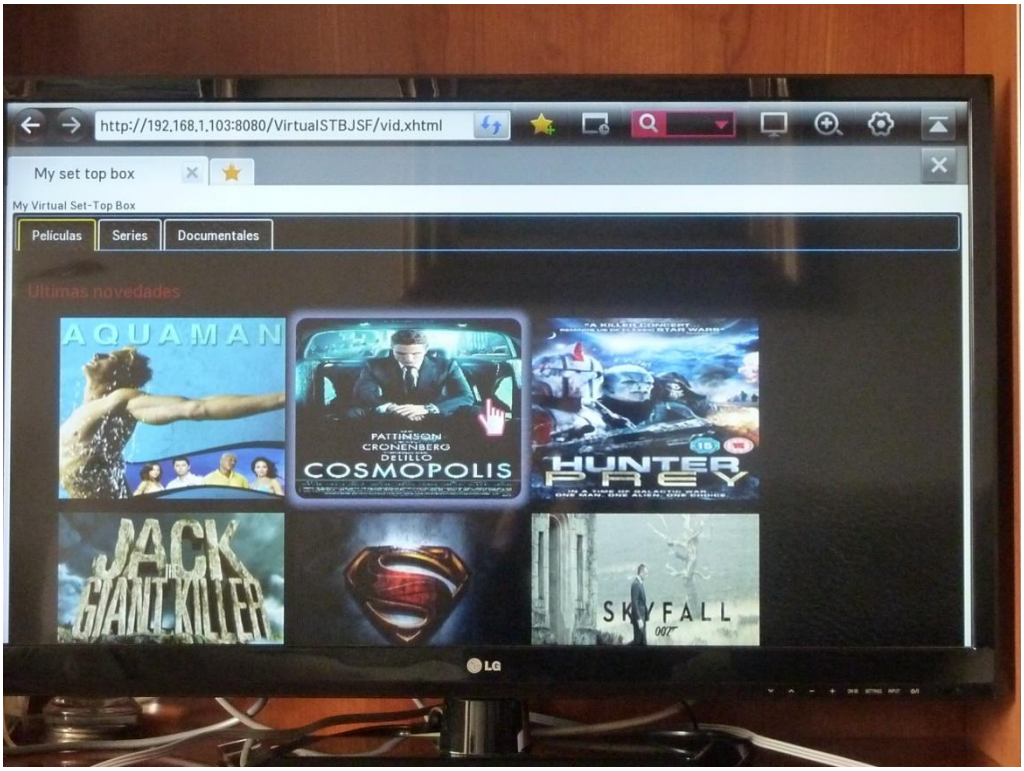

Figura 27: LG Smart TV accediendo al interfaz del STB virtual

La funcionalidad era buena, y la imagen nítida.

6.4 Prueba con HLS

Pese a que en el capítulo 5 hicimos un estudio de compatibilidades y se determinó que no era compatible con ningún navegador, se realiza una prueba para comprobar que efectivamente ningún navegador reproduce el formato HLS.

En cuanto a HLS hacemos dos distinciones, las pruebas de HLS con un VoD a través de una aplicación Android y las de un canal en directo a través del interfaz del STB. A continuación se detallan los resultados de ambas.

6.4.1 *Prueba HLS de un canal en directo a través del navegador*

Añadimos a nuestro interfaz del STB virtual una pestaña para video en directo, en la que pondremos un reproductor de video flash y un video HLS para que al menos uno de ellos pueda ser reproducido. El resultado en un navegador se puede apreciar en la siguiente imagen:

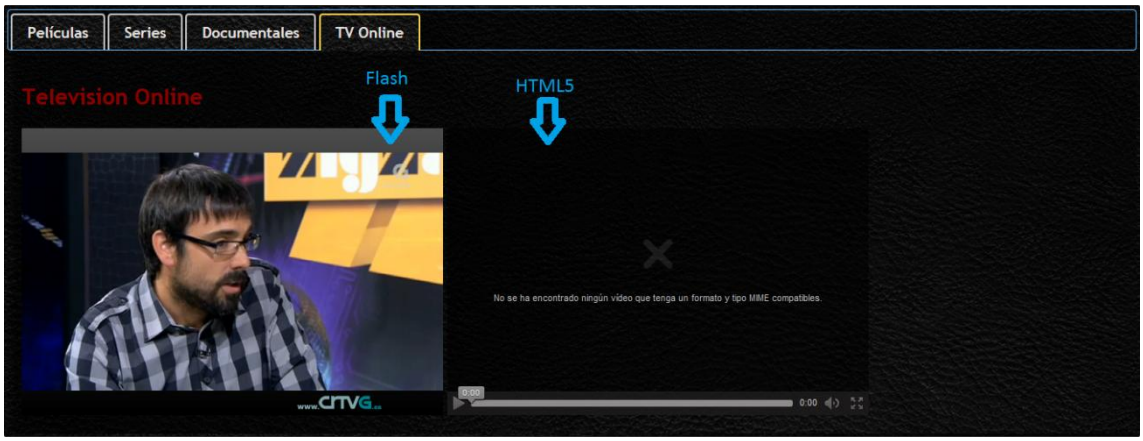

Figura 28: Prueba navegador con HLS

Como se esperaba, el contenido en un navegador Chrome no es compatible con HLS, pero si pudo reproducir el video con flash.

Realizamos también una prueba en una LG Smart TV, el resultado se muestra en la siguiente imagen:

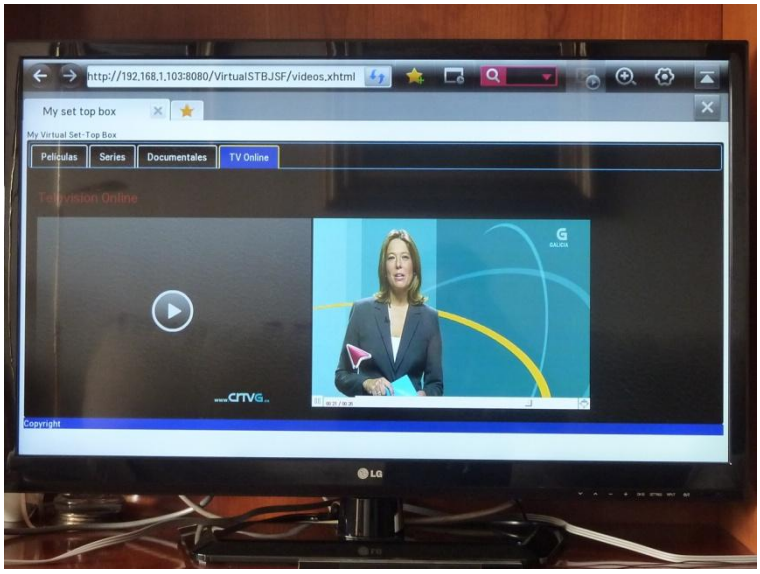

Figura 29: Prueba Smart Tv con HLS

Como se puede ver, el video HLS (a la derecha de la pantalla) se pudo reproducir adecuadamente en el Televisor, lo cual es un resultado inesperado.

A continuación vamos a sintetizar estas pruebas, así como otras realizadas para comprobar su compatibilidad:

Prueba realizada	Resultado	Observaciones
Acceso desde distintos navegadores: Firefox 24, Chrome 29, IE10	OK	En ninguno los navegadores se pudo mostrar adecuadamente el contenido.
Smartphone con Android 4.1	NO OK	A la hora de visulizar el contenido con Chrome com cabía esperar no fue capaz de visualizarlo.
LG Smart TV	OK	El contenido se puede visualizar rapidamente, el contenido se emite con buena calidad y sin cortes. Se comprobó que en pantalla completa tambien funciona correctamente.

Tabla 5: Pruebas HLS

A la vista de las pruebas, destacamos que es posible utilizar HLS para ver un canal en directo, cuando se acceder directamente al interfaz del STB virtual desde un navegador.

6.4.2 Prueba HLS con VoD

Además de las pruebas anteriores, se programa una aplicación en Android para ver si ofrece alguna compatibilidad adicional a las anteriores.

Pese a que el emulador no fue capaz de reproducir el formato HLS, se verificó que con un dispositivo físico con Android 4.1, sí fue capaz, como podemos comprobar en la siguiente captura de pantalla:

Figura 30: Reproducción de video HLS

El video que se reprodujo está formado por varios fragmentos pero tiene una duración determinada.

A la vista de los resultados no podemos concluir que no se pueda reproducir HLS en Google TV, ya que al igual que en el caso de Android 4.1 es posible que si se probara en un dispositivo real funcione adecuadamente.

6.5 Integración de la web en una aplicación

Como cabía esperar las pruebas realizadas en el emulador de Google TV tampoco funcionaron. No es una prueba concluyente el hecho de que no funcione. Alternativamente hemos realizado la prueba sobre un emulador Android 4.3 y un dispositivo físico con Android 4.1. A continuación se muestran los resultados.

6.5.1 Emulador Android 4.3

Se eligió la versión Android 4.3 porque es la última versión que el SDK permite desarrollar. Se usó la aplicación detallada en el capitulo anterior. La figura siguiente muestra el resultado:

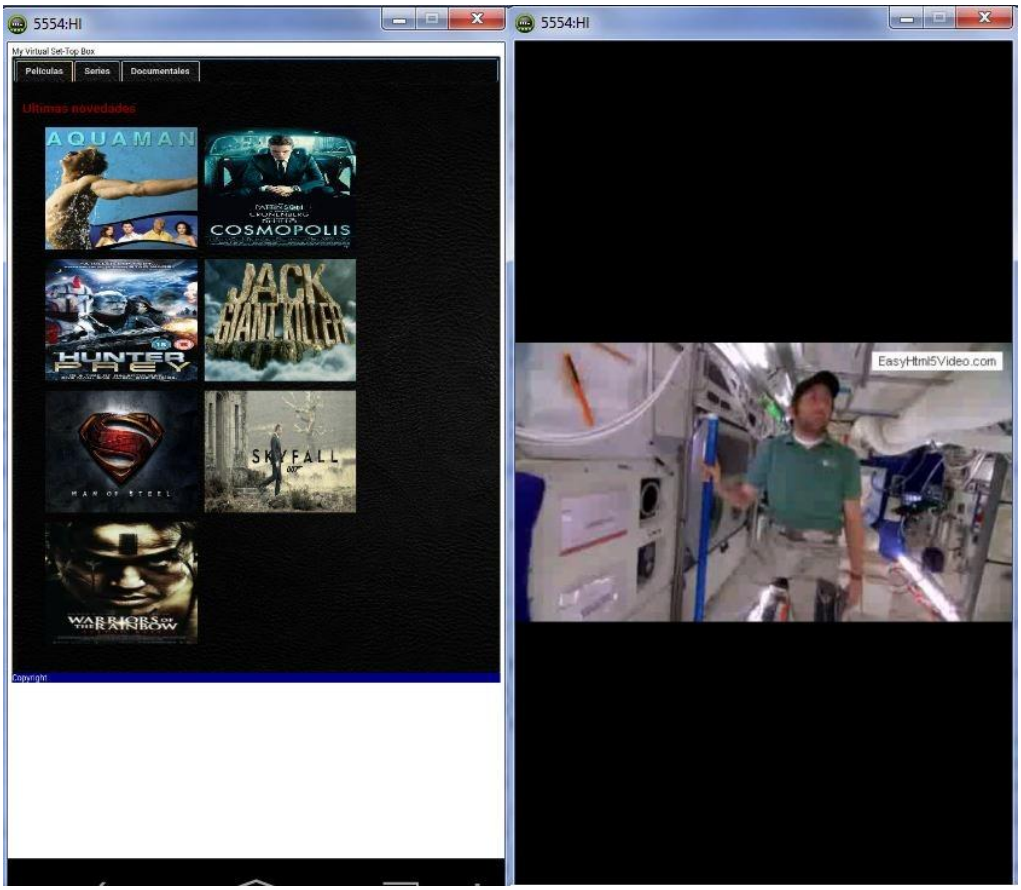

Figura 31: App en emulador Android 4.3

Como se puede observar en la figura anterior la visión es muy parecida a la del navegador, nótese que la parte inferior en blanco es debida a que el contenido no se adapta del todo bien, para la siguiente prueba hicimos una adaptación en una columna del contenido, para comprobar si mejoraba su presentación.

Debido a que esta es la prueba más representativa hemos hecho un análisis de las funcionalidades de la aplicación y si hay algún elemento que tiene un comportamiento anómalo.

Prueba realizada	Resultado	Observaciones
Interfaz	OK	La apariencia de la web es el mismo que en un browser, pero el margen inferior no queda bien fijado
Jquery	OK	Se puede cambiar entre las diferentes pestañas del tab.
Video	OK	El video no cuadra con la pantalla, pero tras hacer doble click se ajusta y se puede reproducir.

Tabla 6: Funcionamiento App en emulador Android 4.3

Como se menciona en la tabla hay algún problema como que cuando se abre el video se muestra demasiado grande y posteriormente hay que dar un doble click para que vuelva a su posición normal. Esto se solucionó introduciendo la siguiente línea:

```
<meta name="viewport" content="width=device-width,initial-scale=1.0,user-scalable=no"/>
```

La cual consiguió que se ajustara a la anchura del dispositivo.

6.5.2 Smartphone con Android 4.1

Para la prueba definitiva se usa un Smartphone Android 4.1, como se menciona en el apartado anterior se cambia la visión para que aparezca el contenido ordenado en una sola columna. El resultado lo podemos ver en la siguiente imagen:

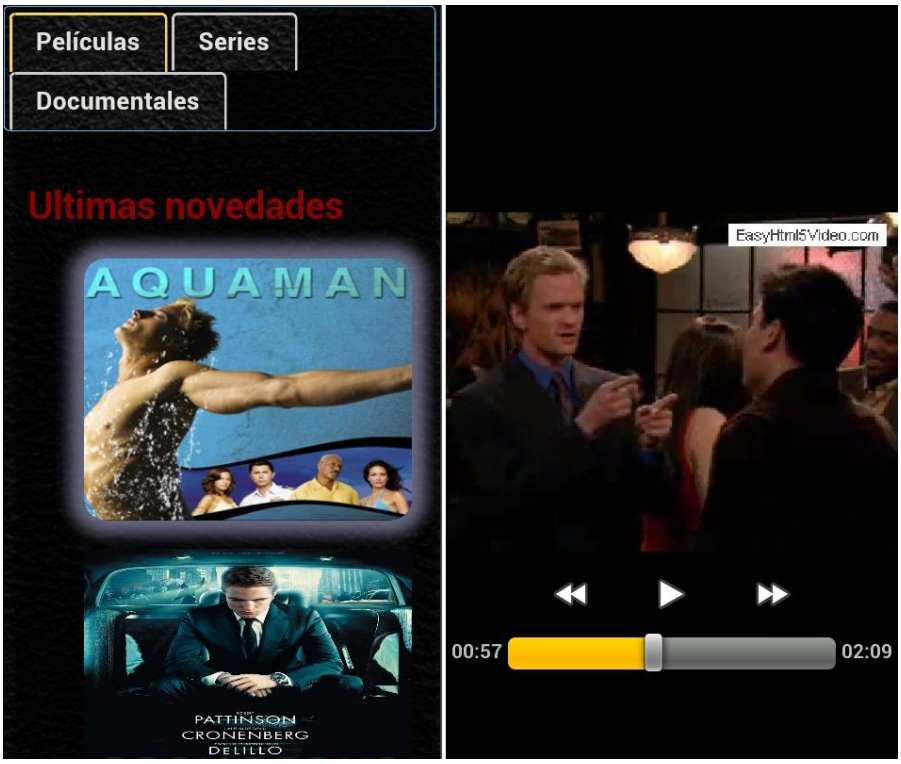

Figura 32: App para Smartphone con Android 4.1

Como se puede ver el interfaz es mucho más amigable que en el caso del emulador.

Los resultados de la prueba de funcionalidad para este caso fueron:

Prueba realizada	Resultado	Observaciones
Interfaz	OK	La apariencia de la web es amigable y está correctamente adaptada al dispositivo.
Jquery	OK	Se puede cambiar entre las diferentes pestañas del tab.
Video	OK	El video se ajusta perfectamente en la pantalla y se puede reproducir en pantalla completa.

Tabla 7: funcionamiento Smartphone con Android 4.1

Debido al buen rendimiento que ha dado la aplicación en dispositivos Android, y a que Google TV, es un Add-on de una versión de Android. Podemos concluir sin equivocarnos que la aplicación funcionará en Google TV.

Capítulo 7

Historia del proyecto

Este proyecto se desarrolla en un periodo de tiempo comprendido entre Octubre de 2012 a Octubre de 2013.

Durante todos estos meses han ido surgiendo problemas y dificultades, a los largo de las diversas etapas en las que se ha dividido este proyecto, todas ellas son objeto del actual capítulo.

Este proyecto es de carácter teórico pese a que se realizan una serie de desarrollos y pruebas para poder estudiar la viabilidad del proyecto. Por este motivo gran cantidad del tiempo empleado en su realización recae en la lectura de papers y trabajos relativos a la temática del proyecto para poder hacer un diseño coherente del mismo.

Cabe destacar que las pruebas documentadas en el capítulo 6 se han realizado paralelamente al desarrollo del proyecto.

7.1 Fases del proyecto

7.1.1 Fase I: Definición de requisitos

- **Descripción de tareas realizadas**

  Debido a la necesidad de un cambio en el modelo de negocio de la TV de pago y a que tecnologías como las Smart TV y la nube están surgiendo con fuerza, se definió la idea y arquitectura de la que se debía dotar al proyecto para poder dar una solución que pueda combinar estas dos tecnologías emergentes. A raíz de esto se definieron los siguientes requisitos:

  - Creación de un STB virtual alojado en la nube que sea capaz de interactuar con la televisión.
  - Utilización aplicación sobre el SO de la TV para poder interactuar con el STB virtual.
  - Aprovechar la capacidad de procesamiento actual de las televisiones para poder realizar operaciones adicionales.

- **Resultados**

  Se decidió empezar por la implementación de una aplicación en la que se pueda comprobar que se puede reproducir contenido multimedia. Posteriormente se implementará el STB virtual, así como el resto de los requisitos.

7.1.2 Fase II: Instalación y familiarización con Google TV

- **Descripción de tareas realizadas**

  El objetivo fue realizar una sencilla aplicación en Google TV que pueda reproducir videos alojados en un servidor web, concretamente un servidor Apache, para simular el acceso a un proveedor de contenidos IP.

  Para ello, primero se estableció que se debía implementar usando HTML5, este nuevo estándar proporciona un tag específico para videos, que es idóneo para nuestra aplicación, la cual además puede especificar diversas fuentes del video de forma sencilla. Para poder programar en HTML5 se tuvo que realizar un tutorial para poder conocer sus nuevas funcionalidades respecto al estándar anterior.

- **Problemas encontrados**

  El emulador de Google TV estaba en fase de desarrollo por aquel entonces y su instalación y correcto funcionamiento fue bastante tedioso, debido a que no había mucha más documentación que la facilitada por Google.

- **Resultados**

  Como se puede comprobar en el capítulo 6, la prueba se realizó exitosamente, el video pudo ser reproducido en el emulador de Google TV.

7.1.3 Fase III: Elección y desarrollo del servidor

- **Descripción de tareas realizadas**

  Tras valorar diferentes opciones se optó por un servidor Glashfish el cuál que ofrece un gran rendimiento y además ofrece una gran integración con Eclipse que fue el entorno utilizado para el desarrollo.

  En cuanto a la tecnología del servidor se eligió JSF, que utiliza el patrón de diseño MVC, que además de simplificar la implementación de las aplicaciones web, debido a su diferenciación en diferentes módulos es altamente reutilizable.

  Para poder realizar la aplicación objetivo tuve que realizar tutoriales de CSS3 y Javascript para poder realizar un interfaz atractivo para el usuario.

  El objetivo era proporcionar el acceso a VoD y Streaming en directo.

- **Problemas encontrados**

  Uno de los principales impedimentos fue a la hora de agregar un streaming de video en directo con HTML5, fue que el estándar de video en directo que se decidió utilizar, HLS, actualmente no se puede reproducir en ningún navegador web.

- **Resultados**

  Se creó el interfaz del STB, este interfaz fue capaz de reproducir videos del servidor web del que se hablo en la fase anterior, además se le dotó de lógica que le permite presentar y reproducir los videos que se alojan en la web independientemente de que se borren y añadan nuevos en cualquier momento.

7.1.4 Fase IV: Estudio de compatibilidad entre STB y Google TV

- **Descripción de tareas realizadas**

  Se realiza una serie de estudio de investigación acerca de como poder compatibilizar estas tecnologías, también se creará una aplicación para poder reproducir el contenido de la web en el emulador Google TV.

- **Problemas encontrados**

  Google TV es un Add-on sobre la versión de Android 3.2 la cual presenta incompatibilidades añadidas con videos y sus formatos.

  Estas incompatibilidades se explican ampliamente en el capítulo 6.

- **Resultados**

  En esta fase se comprueba que la aplicación no funciona en el emulador de Google TV, adicionalmente se verifica que si funciona con el emulador Android 4.3 yen un Smartphone con Android 4.1. Este hecho hace pensar que si funcionará sobre las Smart TV actuales, muchas de las cuales llevan un SO Android 4.x.

7.1.5 Fase V: Búsqueda de alternativas y viabilidad del proyecto

- **Descripción de tareas realizadas**

  En esta fase se centraron los esfuerzos en poder ofrecer alternativas a los problemas encontrados y comprobar si son compatibles con Google TV.

  También se explora otras opciones, como su adaptación a dispositivos móviles.

- **Resultados**

  Aunque bien es cierto que el emulador de Google TV no es compatible con la televisión en directo y HTML5 se pudo comprobar que en versiones posteriores de Android sí son compatibles.

7.1.6 Fase VI: Documentación

- **Descripción de tareas realizadas**

Esta fase consiste en la documentación del trabajo realizado en las anteriores fases del proyecto y de la realización de esta memoria.

7.2 Resumen

Como complemento a la información detallada de todas las fases se va a facilitar una tabla con la duración de cada una de las fases.

Como se puede apreciar en la tabla 8, hay dos fases que han requerido una gran duración.

Por un lado la fase III, coincidió con la preparación y realización de exámenes de la universidad lo que hizo que se ralentizara el ritmo de trabajo, aunque tampoco son excesivos meses debido a que es la fase de más dificultad del proyecto.

Otra fase afectada fue la fase IV duró tres meses cuando a priori tampoco debería ser una tarea tan extensa, pese a que ha requerido muchas horas de búsqueda de información en internet y de pruebas para verificar o descartar su funcionamiento, esto es debido a que empecé a realizar prácticas en empresa en Abril del 2013 y he tenido menos tiempo para poder realizar el proyecto desde Abril hasta Septiembre.

Fase	Descripción	Duración(mes)
Fase I	Definición de requisitos	0.5
Fase II	Instalación y familiarización con Google TV	1
Fase III	Elección y desarrollo del servidor	4
Fase IV	Estudio de compatibilidad entre STB y Google TV	3
Fase V	Búsqueda de alternativas y viabilidad del proyecto	1
Fase VI	Documentación	1

Tabla 8: Fases del proyecto

Capítulo 8

Conclusiones y trabajos futuros

8.1 Conclusiones

En los últimos años la evolución de la electrónica de consumo ha propiciado que las televisiones hayan dejado de ser dispositivos pasivos para convertirse en dispositivos interactivos. Estos dispositivos están dotados de un SO que les permite una fácil interacción con el usuario y les proporciona conectividad a internet, son las denominadas Smart TV.

Paralelamente, se ha experimentado en la sociedad una creciente demanda de usuarios que acceden a contenido multimedia a través de internet. Este hecho ha impulsado a los proveedores de IPTV y a las aplicaciones OTT, las cuales son muy demandadas debido a su flexibilidad y facilidad de acceso al contenido.

Al mismo tiempo, aprovechando el éxito de las soluciones Cloud Computing, han surgido varias aplicaciones OTT que aprovechan la capacidad de esta tecnología para ofrecer una forma eficaz de reproducir y compartir el contenido multimedia en cualquier dispositivo.

Hasta el presente, era necesario disponer de un STB para acceder a los contenidos de un proveedor IPTV, que debía ser previamente contratado. Este proyecto ha propuesto el concepto y ha introducido el diseño de un STB virtual, el cual estaría alojado en la nube. El STB virtual puede proporcionar las mismas funcionalidades que un STB tradicional. Para ello, utiliza unos plug-in que son los encargados de comunicarse con el proveedor IPTV y garantizar su correcta accesibilidad.

En este modelo se elimina la necesidad de un hardware físico adicional y de una subscripción previa para poder acceder a visualizar el contenido de un proveedor IPTV. Además el usuario tendrá una visión global de todo el contenido de los distintos proveedores IPTV en una misma aplicación.

Tras proponer el modelo del STB virtual se han realizado diferentes estudios de compatibilidad y una prueba de desarrollo del mismo. Las conclusiones derivadas de estos estudios son que no hay un impedimento tecnológico para la implementación de ese modelo, debido a las múltiples alternativas que hay en el estado del arte actual. Por lo cual, es viable su implementación para uso real en un futuro.

No obstante, en cuanto a las pruebas realizadas sobre Google TV, hay que mencionar que Google TV ha sido uno de los grandes fracasos de Google. Esta tecnología no ha sido capaz de implantarse en el mercado, debido al alto precio de los STBs que disponían de esta tecnología y a que los fabricantes de televisiones optaron por sus propias soluciones propietarias. En la actualidad se está tendiendo hacia Android TV. La diferencia entre ambas es que Android TV engloba cualquier TV que tenga un SO Android, ya sea interno o externo, mientras que Google TV es un Add-on de la versión Android 3.2, que se usa mayoritariamente en tablets. Por otra parte, la mayoría de los SO propietarios de las Smart TV soportan HTML5 y además los fabricantes de televisiones, están incorporando las últimas versiones de Android en sus televisiones de más alta gama. En consecuencia, se concluye, que las Smart TV son compatibles con este modelo, y en un futuro cercano se prevé que la convergencia hacia un SO único haga más fácil su implementación.

Por otra parte, centrándonos en los objetivos marcados al inicio del proyecto, se ha conseguido realizar un diseño realista y portable de un STB virtual en el cual hemos definido diferentes escenarios implementables. No obstante, es una solución abierta en la que se podrían implementar otros escenarios.

También se ha logrado implementar un interfaz para el STB virtual, que pueda sea compatible con la mayoría de los navegadores y dispositivos. Esta implementación ha tenido en cuenta que el recepto final será una TV, Smartphone o Tablet, facilitando su exportación a un SO.

Además se ha conseguido realizar una aplicación que logre visualizar el contenido multimedia, alojado en un servidor externo, simulando el proveedor IPTV, con lo que se puede concluir que es exportable a un escenario real.

En lo relativo a la visualización de canales en directo, se ha podido reproducir tanto en dispositivos Android como en una Smart TV, lo cual significa que la prueba fue exitosa.

Por último cabe añadir que se han cumplido los objetivos marcados para el proyecto. Aunque no se ha podido comprobar su funcionalidad en una televisión Google TV, las pruebas hacen indicar que el resultado sería satisfactorio.

8.2 Líneas futuras

En este apartado se presentan algunas de las mejoras y trabajos futuros que se pueden realizar sobre este proyecto. A continuación se va a presentar algunas de las líneas de investigación futuras.

- **Implementación de los sistemas de seguridad**

  Debido a la falta de tiempo, la implementación de sistemas de seguridad quedó fuera del alcance de este proyecto. No obstante es una de las líneas más importantes a continuar si se desea seguir con el proyecto. Ningún proveedor adquiriría el sistema si el contenido no viaja seguro a través de internet y no se tienen en cuenta el DRM.

- **Investigar y desarrollar más funciones de los plug-in en nuestro sistema**

  En nuestro proyecto se ha desarrollado un sistema de plug-in en local, los cuales son informados cuando el usuario realiza cualquier acción y evalúan si tienen que actuar.

  El sistema de plug-in aporta múltiples funcionalidades, una de los posibles desarrollos futuros es el de integrar varias EPGs provenientes de distintos proveedores IPTV en una sola EPG.

- **Utilización de DASH**

  Para el proyecto se ha utilizado HLS como protocolo para emitir y recibir video en streaming. Como hemos concluido, este formato es un estándar propietario de Apple, por lo que no es compatible con la mayoría de los navegadores.

  DASH es un estándar que soporta tanto VoD como streaming en directo en múltiples formatos como MPEG4 y MPEG-2 TS, tiene en cuenta varios escenarios de DRM y soporta SVC.

- **Pruebas**

  Durante el proyecto, no se ha podido realizar una prueba sobre un dispositivo físico con el Add-on de Google TV, esto es debido a la imposibilidad económica de adquisición de un televisor o STB con Google TV. Por ello, esta prueba se concibe como una de las que se implementarían de forma más fácil en un futuro.

  Adicionalmente se podrían realizar otras pruebas, como verificar la correcta integración del sistema con otros sistemas operativos. A priori el más sencillo de implementar sería bajo iOS. Esto es debido a que HLS es un sistema propietario de Apple y este funcionará adecuadamente en estos dispositivos.

Apéndice A

A. Presupuesto

En este apartado se va a detallar el presupuesto del proyecto. En el presupuesto se incluyen por separado tanto el coste de personal así como de los materiales utilizados. La suma de ambos constituirá el coste total del proyecto.

1. Costes de personal

La duración del proyecto ha sido de 12 meses. En cuanto a la dedicación diaria, ha fluctuado mucho debido a periodos de exámenes y a que trabaje 6 meses a media jornada. Se estima que la dedicación diaria en media, es media jornada, 4 horas. Una dedicación de 4 horas al día durante 12 meses, hace un total de 960 horas.

En cuanto a los honorarios de un ingeniero, estos responden a acuerdos de libre mercado en base a acuerdos comerciales entre particulares y profesionales. En el pasado, el Colegio Oficial de Ingenieros de Telecomunicaciones publicaba a modo orientativo los honorarios que debía cobrar un ingeniero por hora. Con la nueva normativa europea, los colegios han dejado de publicar dicha información, con lo que para el cómputo del coste humano del proyecto no podemos usar una información actualizada. Tomaremos una referencia anterior para dicho cálculo en la que se estipulaba un coste de 75 euros por hora de ingeniero. Por otra parte, el coste de un director de proyecto se estima en un 8% del coste total del proyecto. En este caso existirá un director, contabilizando así el tiempo dedicado por parte del tutor asignado por parte de la universidad.

La siguiente tabla resume los costes debido a los honorarios de las partes descritas.

Concepto	Coste	Cantidad	Total
Ingeniero de Proyecto	75 € / Hora	960 horas	72.000 €
Director de Proyecto	8%	72.000 €	5.760 €
Total sin i.v.a			77.760 €
Total con i.v.a(21%)			94.090 €

Tabla 9: Costes de personal

2. Costes de material

En cuanto al coste de material, este se refiere al material usado tanto para el estudio y desarrollo de la memoria así como de las instalaciones donde se ha desarrollado

2.1Equipo informático

En este apartado incluimos el portátil utilizado a lo largo del proyecto, el móvil con el cual se han realizado las diferentes pruebas y el gasto asociado al mantenimiento del mismo.

Concepto	Coste
Material Informático: • Portatil Toshiba	550 €
Dispositivos móviles: • Huawei Y 300 Tarjeta Yoigo tarifa 1	253 € 9 € /mes
Total Proyecto	**803 €**

Tabla 10: Equipo informático

2.2Licencias Software

El sistema operativo instalado en el equipo informático en el que se ha realizado el proyecto es Windows 7 Professional de Microsoft. Además es necesario otros paquetes de software: Visual Studio Professional, Team Foundation Server, Microsoft Office 2010. El coste de este software es de 2298 euros al año. La depreciación que sufren los sistemas y programas informáticos según el Real decreto 1777 de 2004 es de un 33 % por año. Por lo tanto se incluirán Se incluirán por tanto 758.34 euros en los costes del proyecto en concepto de licencias de Software.

Concepto	Coste	Cantidad	Total
Equipo Informático	803 €	33,3 % depreciación	267,64 €
Licencias Software	2.298 €	33,3 % depreciación	758,34
Total sin i.v.a			1.026 €
Total con i.v.a(21%)			1.241 €

Tabla 11: Costes de material

3 Presupuesto completo

Teniendo en cuenta lo descrito anteriormente, el precio total para el proyecto completo se resume en la siguiente tabla.

Concepto	Coste
Total Honorarios	94.089,60 €
Total Material	1.241,44 €
Total	95.331,04 €

Tabla 12: Coste total

Apéndice B

B. Glosario de términos

CAM: Conditional Access Module.
CI: Common Interface.
CPCM: Content Protection & Copy Management
CSS: Cascading Style Sheets.
CW: Control Word
DASH: Dynamic Adaptive Streaming over HTTP.
DLNA: Digital Living Network Alliance
DRM: Digital Right Management.
DVB: Digital Video Broadcasting
ECM: entitlement control message
EMM: entitlement management message
ETSI: Instituto Europeo de Estándares de Telecomunicaciones
HDS: HTTP Dynamic Streaming.
HEVC: High Efficiency Video Coding.
HKMS: Home Key Management System
HLS: HTTP Live Streaming.
HTML: HyperText Markup Language.
IMS: IP Multimedia Subsystem
IPTV: Internet Protocol Television.
JSF: JavaServer Faces
JSON: JavaScript Object Notation.
MBMS: Multimedia Broadcast Multicast Services
MPEG: Moving Picture Experts Group
MVC: Modelo Vista Controlador.
OIPF: Open IPtv Forum
OTT: Over The Top
PES: Packetized Elementary Stream
QoS: Quality of Service
REST: Representational State Transfer.
RTMPe: Real Time Messaging Protocol encrypt
SaC: Secure Authenticated Channel
SK: Service Key
SO: Sistema Operativo
SOAP: Simple Object Access Protocol.
STB:Set Top Box.
SVC: Scalable Video Coding.
TDT: Televisión Digital Terrestre
TISPAN: Telecommunications and Internet converged Services and Protocols for Advanced Networking
TS: Transport Stream

TV: Television.
UE: Unión Europea.
UPnP: Universal Plug and Play.
VoD: Video On Demand.

Bibliografía

[Aponte09] Juan Carlos Aponte Gonzalez, "Set Top Box DVB-T", Available at http://www.actuonda.com/pdf/presentaciones_seminario/Seminario_TV_Digital_Bogota_STB_Network_Broadcast.pdf, Jul 2009

[App11] Apple TV Website, Available at http://www.apple.com/appletv/, December 2011.

[Box11] Boxee box, Available at http://www.boxee.tv/, April 2011

[Dashiso12] ISO/IEC 23009-1 "Dynamic adaptive streaming over HTTP (DASH)" : http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=57623

[Diaz09] Díaz-Sánchez, D.; Sanvido, F.; Proserpio, D.; Marin, A., "DLNA, DVB-CA and DVB-CPCM integration for commercial content management," Consumer Electronics, IEEE Transactions on , vol.56, no.1, pp.79,87, February 2010

[Diaz10] Díaz-Sánchez, D.; Sanvido, F.; Proserpio, D.; Marin, A., "DLNA, DVB-CA and DVB-CPCM integration for commercial content management," Consumer Electronics, IEEE Transactions on , vol.56, no.1, pp.79,87, February 2010

[Diaz11] D. Díaz-Sánchez, F. Almenarez, A. Marín, D. Proserpio, P. Arias Cabarcos, “Media Cloud: an Open Cloud Computing Middleware for Content Management," IEEE Trans. Consumer Electron., vol. 57, no. 2, May 2011.

[DmCloud] Dailymotion Cloud , Available at https://www.dmcloud.net/en/, Feb 2008

[Dro11] DropBox, Available at https://www.dropbox.com, December 2011

[Goo11] Google TV Overview, Available at http://www.google.com/tv/, December 2011.

[Graft13] M.Grafl,C.Timmerer,H.Hellwagner,G.Xilouris,G.Gardikis,D.Renzi,S.Battista,E.BorcociandD.Negru,“Scalable Media CodingEnabling Content-Aware Networking,”IEEE Multimedia, 2013

[Hulu09] Hulu, Available at http://www.hulu.com/, Julio 2009

[ICl11] iCloud, Available at http://www.icloud, December 2011.

[Inx11] inXtron's Personal Cloud Server, Available at http://www.inxtron.com/resources/articles/personal-cloud-server, December 2011.

[Lg11] Simply Smarter, LG Smart TVs. Available at http://infinia.lge.com/archives/2108, December 2011.

[Lhbeststb] Best Set Top Box, Avaliable at http://lifehacker.com/5946193/how-to-find-the-best-tv-set-top-box-and-ditch-cable-once-and-for-all

[Mpeg2wikitel] Ru Torres, http://wikitel.info/wiki/Imagen:Flujose%C3%B1almpeg2.jpg

[Navarro06] Rafael Navarro Marset,"REST vs Web Services",2006

[Netflix11] Netflix Management. Available at https://signup.netflix.com/MediaCenter?id=5380&country=1&rdirfdc=true#rhastings, December 2011

[NowTV12] NowTV, Available at http://www.nowtv.com/, Julio 2012

[Pan11] Smart networking, Available at http://panasonic.net/avc/viera/popup/smart_networking, December 2011.

[Pog11] Pogoplug Multimedia Sharing Device, Available at http://www.amazon.com/Pogoplug-POGO-E02-Multimedia-Sharing-Device/dp/B0033WSDR4, December 2011-

[Roku10] Roku Website, Available at http://www.roku.com/, September 2010

[Sam11] Smart TV with Samsung Apps, Available at http://www.samsung.com/us/article/apps-built-for-your-tv, December 2011.

[Schwarz07] H. Schwarz, D. Marpe and T. Wiegand, "Overview of the Scalable Video Coding Extension of the H.264/AVC Standard", IEEE Transactions on Circuits and Systems for Video Technology, vol. 17, no. 9, pp. 1103-1120, September 2007

[Son11] Sony redefines home entertainment with the world's first HDTV powered by Google TV, Retreived at December 2011 http://news.sel.sony.com/en/press_room/consumer/television/release/58663.html.

[Startcapps] Startcapps, "Web Services – REST vs SOAP", Available at http://www.startcapps.com/blog/web-services-rest-vs-soap/

[StrmNat08] Stream Nation, Available at https://www.streamnation.com/ , 2008

[Sullivan12] Sullivan; J.-R. Ohm; W.-J. Han; T. Wiegand (2012-05-25)."Overview of the High Efficiency Video Coding (HEVC) Standard" (PDF). IEEE Transactions on Circuits and Systems for Video Technology. Retrieved 2012-09-14.

[Sundareshan09] B. Sundareshan, "Digital Set Top Box (STB) – Open Architecture/Interoperability Issues",2009

[tandroidHTML5] HTML5WebView layout, Availale at http://www.tandroid.org/html5webview

[Ton11] Tonido-Run your own personal cloud, Available at http://www.tonido.com/ , December 2011

[Tumaes] Tu maesto web, "¿Que es MVC ?", Available at http://www.tumaestroweb.com/curiosidades/que-es-mvc/

[Vetro07] Anthony Vetro, Charilaos Christopoulos, and Huifang Sun. Video transcoding architectures and techniques: an overview. IEEE Signal Processing Magazine, pp. 18–29, March 2003.

[Viddler08] Viddler, Available at https://www.viddler.com/plans, Abril 2008

[Vzaar08] Vzaar, Available at http://vzaar.com/features, 2008

[Wherever11] Wherever, Available at http://www.wherever.tv/index2.jsf, 2011

[Zum11] Personal Cloud Computing With a Twist – ZumoDrive, http://lonewolflibrarian.wordpress.com/2009/05/16/personal-cloudcomputing-with-a-twist-zumodrive-05-16-09/